EFFECTS OF MEASUREMENT ERRORS ON PARAMETER ESTIMATION VIA NEURAL NETWORKS

By

Srinivasa Rao Chintala

DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

DECEMBER, 1998

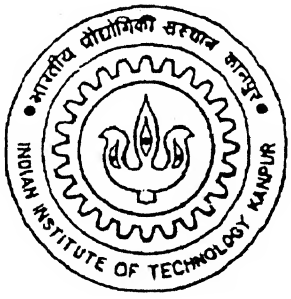
EFFECTS OF MEASUREMENT ERRORS ON PARAMETER ESTIMATION VIA NEURAL NETWORKS

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Master of Technology

by

Srinivasa Rao Chintala



to the

Department of Aerospace Engineering
Indian Institute of Technology
Kanpur

December, 1998

CERTIFICATE

It is certified that the work contained in the thesis entitled, "**Effects of Measurement Errors on Parameter Estimation via Neural Networks**", by *Srinivasa Rao Chintala*, has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Dr. S. C. Raisinghani
Professor
Department of Aerospace Engg.
I.I.T., Kanpur.

Dr. A. K. Ghosh
Department of Aerospace Engg.
I.I.T., Kanpur.

December, 1998.

Acknowledgements

I wish to record my deep sense of gratitude to my esteemed teachers and thesis supervisors, **Dr. S.C. Raisinghani** and **Dr. A.K. Ghosh** for their meticulous guidance, invaluable suggestions, constructive criticism and constant encouragement during the tenure of this thesis work. I shall always remain indebted to them for the precious time they have spared for me and the patience with which they always enlightened my path.

With all sincerity, I thank **Dr. P.K. Kalra**, the professor of Electrical Engg. Deptt., for his guidance in understanding Neural Networks and its applications.

I express my special thanks to my friends Mr. G.V.S. Bhaskar Reddy, Mr. K. Janardhana Reddy and Mr. P. Narayana Rao for their tremendous help in bringing this work to the present form.

No words can be coined to express my deep sense of gratitude to my parents and my youngers who have been a constant source of inspiration to me.

Finally, I wish to acknowledge all the friends and faculties, who were a part of the process of intellectual enrichment and personal growth in many aspects of my life, in this year and a half stay at IIT, Kanpur.

Srinivasa Rao Chintala
IIT, Kanpur.

Abstract

Recently, two methods based on feed forward neural networks, christened "the Delta method" and "the Zero method", have been proposed for estimating aircraft stability and control derivatives (parameters) from flight data. The methods have been validated on simulated as well as real flight data wherein the flight data used were either without any measurement errors, or were corrected for the measurement errors before their use for parameter estimation. In the present work, scope of both the Delta and the Zero methods is expanded by investigating their ability to extract parameters from raw sensor data without prior corrections for possible measurement errors like the bias, scale factor, measurement noise, etc. Results are presented for simulated data with varying amounts of measurement errors. It is shown that the Delta method can be applied directly to the recorded flight data that may have errors of bias, scale factor, and measurement noise. However, the recorded data needs to be corrected for the time shift errors. Finally, real flight data is used with and without correcting it for the bias and scale factor, and the resulting parameter estimates via the Delta method are shown to compare well with the estimates via the maximum likelihood method.

Contents

List of Figures

List of Tables

List of Symbols

Symbols:

a_y, a_z	accelerations along y and z body axes, m/s^2
C_l	rolling moment coefficient
C_n	yawing moment coefficient
C_y	force coefficient along Y-axis
C_{l0}, C_{n0}, C_{y0}	aerodynamic trim parameters
b	wing span, m
c	mean aerodynamic chord, m
g	acceleration due to gravity, m/s^2
I_x, I_y, I_z	moments of inertia about x, y, and z axis, $kg\text{-}m^2$
I_{xy}, I_{yz}, I_{zx}	cross products of inertia, $kg\text{-}m^2$
k_β	scale factor for angle of side-slip
m	aircraft mass, kg
p, q, r	roll, pitch and yaw rates, rad/s
p_b, r_b	biases in roll and yaw rates, rad/s
$\overline{q}$	dynamic pressure, N/m^2
s	half the wing span, m
S	reference wing area, m^2

u, v, w	velocity components along x, y and z body axes, m/s
W_f	fuel weight (mass), kg
W_{ij}	connection weight
X_a, Y_a, Z_a	accelerometer off-set distances relative to C.G., m
X_n, Y_n, Z_n	flight log off-set distances relative to C.G., m
$\Delta X_{C.G.}$, $\Delta Y_{C.G.}$, $\Delta Z_{C.G.}$	C.G. location relative to reference point R, m
α	angle of attack, rad
β	angle of side-slip, rad
β_b	bias in angle of side-slip, rad
δa	aileron deflection, rad
δr	rudder deflection, rad
δ_f	flap deflection, deg
λ	logistic gain factor
μ	learning rate factor
Ω	momentum rate parameter
ρ	density of air
Δ	small perturbation

Superscripts:

e	Referred to experimental axes
R	Referred to reference point
.	derivative with respect to time

Abbreviations

ANN	Artificial Neural Network
ATTAS	Advanced Technologies Testing Aircraft System
BPA	Back Propagation Algorithm
C.G.	Center of Gravity
DLR	Deutsche Forschungsanstalt fur Luft-und Raumfahrt
FFNN	Feed Forward Neural Network
IAS	Indicated Air Speed
ML	Maximum Likelihood
MSE	Mean Square Error
SP	Special Position (of landing flap)

Stability and Control Derivatives

$$C_{lp} = \frac{\partial C_l}{\partial(\frac{pb}{2V})}, \quad C_{np} = \frac{\partial C_n}{\partial(\frac{pb}{2V})}, \quad C_{yp} = \frac{\partial C_y}{\partial(\frac{pb}{2V})},$$

$$C_{lr} = \frac{\partial C_l}{\partial(\frac{rb}{2V})}, \quad C_{nr} = \frac{\partial C_n}{\partial(\frac{rb}{2V})}, \quad C_{yr} = \frac{\partial C_y}{\partial(\frac{rb}{2V})},$$

$$C_{l\beta} = \frac{\partial C_l}{\partial \beta}, \quad C_{n\beta} = \frac{\partial C_n}{\partial \beta}, \quad C_{y\beta} = \frac{\partial C_y}{\partial \beta},$$

$$C_{l\delta a} = \frac{\partial C_l}{\partial \delta a}, \quad C_{n\delta a} = \frac{\partial C_n}{\partial \delta a}, \quad C_{y\delta a} = \frac{\partial C_y}{\partial \delta a},$$

$$C_{l\delta r} = \frac{\partial C_l}{\partial \delta r}, \quad C_{n\delta r} = \frac{\partial C_n}{\partial \delta r}, \quad C_{y\delta r} = \frac{\partial C_y}{\partial \delta r},$$

Chapter 1

Introduction

During the last three decades, the field of system identification has emerged as an important discipline. Application of this field finds a place in variety of disciplines like Aerospace engineering, Chemical engineering, Metallurgical plants, and so on. The general problem of identifying a system based on measurement of input and output variables is defined as obtaining functional relationship between the inputs and outputs.

Among all the subfields of system identification, the most widely applied is that of parameter estimation, wherein an assumed mathematical model based on phenomenological considerations is used to estimate the properties of the dynamic system. The assumed model, however, will not be an exact representation of the system, no matter how careful its selection is. To reduce this difference, the objective is redefined in terms of parameter estimation which attempts to make the mathematical model the best possible representation of the essential characteristics of the system.

One of the successful application of this subfield (parameter estimation) is the estimation of numerical values of aerodynamic stability and control derivatives (aircraft parameters) from flight test data. The reasons behind this success are:

1. the dynamics of an airplane is well understood and can be appropriately modelled
2. better measurement techniques and data processing
3. better estimation algorithms made possible by ever increasing computational power of digital computers.

Three distinct approaches for estimating aircraft parameters are:

1. Theoretical Methods
2. Wind Tunnel Testing
3. Flight Testing

At the initial stages of aircraft design, theoretical methods[1-3] provide the only convenient way of estimating the aircraft parameters. However, the theoretical estimates are not accurate enough for many difficulties. On the other hand, the wind tunnel methods, although more reliable, are time consuming, expensive, and also suffer from discrepancies due to interference effects of support system, wall effects, turbulence level, etc. As a final verification, it is, therefore, desirable that the aircraft parameters be estimated from the actual flight test data.

The most commonly used methods for parameter estimation from flight data are broadly classified into the following categories.

- Equation error methods
- Output error methods (the maximum likelihood method being the most popular)

The principle of least squares is used in the equation error methods, the error gets minimized with respect to the unknown parameters in each of the

equations. Its advantages includes computational simplicity, non-iterative nature and applicability to both linear and nonlinear models. In the output error methods, the error between the measured and the model response produced by identical inputs is minimized. The method assumes that there exists no modelling errors. This method processes the measurement noise while assuming the model representation of the given system to be exact. The maximum likelihood (ML) estimator and its several variants have been the most successfully used methods for estimating parameters from flight data.[4,5]

The above mentioned methods require accurate measurement of input-output variables. This requirement is difficult to meet in practice due to many types of measurement errors present in data. A few typical measurement errors present in data are: the bias (also called zero shift), scale factor, time shift and measurement noise. For conventional parameter estimation methods, it is, therefore, required that the data compatibility check (sometimes referred to as flight path reconstruction) be performed to get accurate results. Thus, the data compatibility check becomes an integral part of the process of parameter estimation.

The data compatibility is performed by using the standard kinematic equations, and applying a Kalman filter or ML algorithm.[6] The compatibility check provides an accurate information about the aircraft states, and also estimation of biases, scale factors and time shifts in the recorded data. Elimination of such errors from flight measurements prior to estimation of parameters helps to improve accuracy of estimates. However, the process of data compatibility check for estimation of errors can itself be quite sensitive to the methodology employed, and at times, it may be as time consuming as the process of estimating parameters itself.

Recently, a new thrust area has emerged in the area of aircraft aerodynamic modeling and parameter estimation: development of techniques using artificial neural networks (ANNs) for flight vehicle identification. The ANNs have been successfully used for modeling in various fields such as signal pro-

cessing, pattern recognition, system identification and control, and so on.

Historically, ANNs were developed out of continued interest in the constructional and functional details of the human brain, and the research thereupon to understand its neurological capabilities such as its ability to learn, to retain and to recall information on demand.

A class of artificial neural networks called the feed forward neural networks (FFNNs) are most widely used among all types of ANNs for aircraft modeling. As plenty of literature is available on FFNNs, here we present only a brief outline about them. The FFNNs have neurons arranged in layers like directed graphs, and these are static in nature. FFNNs lead to a black box model in which no physical significance can be assigned to either the network structure or to the network weights. FFNNs work as general function approximators, and are capable of approximating any continuous function to any desired accuracy by an appropriate network architecture.[7] This ability of FFNNs has been utilized to model aircraft aerodynamics [8-11] successfully.

Very recently, Raisinghani et al [12] have used FFNNs for modeling aircraft aerodynamics wherein measured responses and control inputs are mapped to the aerodynamic coefficients; such a neural model is subsequently used to estimate aircraft parameters by the Delta method and the Zero method proposed in Ref. 12. It is also mentioned that the FFNN modeling does not require any a priori postulation of the mathematical model for the aircraft aerodynamics. Applicability of the Delta method and the Zero method for estimation of lateral-directional parameters from simulated as well as real flight data has been demonstrated, respectively, by Ghosh et al[13] and Raisinghani et al.[14]

However, both the simulated and real flight data analyzed in Refs. 13 and 14 had no measurement errors except for the measurement noise, i.e., no bias, scale factor, or time shift errors were present. For the real flight data, estimates from the data compatibility check for the bias, the scale factor and the time shift were used for correcting the data before being used for parameter

estimation via the Delta method[13] and the Zero method.[14]

The conventional methods like the ML method postulate a model of the aircraft, and define a cost function based on the difference between the model and the aircraft responses for identical control inputs. Because the model response has no measurement errors while the measured response might have bias, scale factor and time shift errors, it becomes necessary to remove such errors from the measured data before the cost function is defined. In contrast, FFNN model does not require an a priori model of the aircraft; it maps the motion and control variables to the total aerodynamic coefficient to yield an aerodynamic model of the aircraft. Such a model is then utilized by the Delta method to estimate parameters. It was, therefore, conjectured that the use of a neural model to estimate parameters via the Delta method might not be affected by the presence of measurement errors in the motion/control variables, and thus permit use of the recorded data directly, i.e., without being first corrected for the measurement errors.

In the present work, both the Delta and the Zero methods are used for extracting parameters from 1) simulated flight data corrupted with bias, scale factor and measurement noise, and 2) from recorded (real) flight data that are not corrected for the bias and the scale factor errors. For both cases, the parameter estimates are compared with the estimates obtained from the corresponding flight data that either do not have the measurement errors (simulated data), or are corrected for the bias and scale factor prior to their use for parameter estimation (real flight data). The results show that the Delta method is capable of estimating parameters from the raw data using the measured values of the motion variables that are not corrected for the bias and/or the scale factor.

Specifically, various aspects of application of FFNN to aircraft lateral-directional aerodynamics, details about the Delta and the Zero methods, and estimation of parameters from simulated and real flight data containing different levels of measurement errors are dealt as follows:

- In Chapter 2, a brief discussion about the FFNN is presented, and the steps involved in the BPA are also given.
- In Chapter 3, a detailed description of the methods used for parameter estimation, namely the Delta method and the Zero Method, is given.
- Chapter 4 described the generation of simulated flight data, preprocessing of real flight data, and the lateral-directional aerodynamic modelling of an aircraft using FFNNs.
- In Chapter 5, results for simulated flight data corrupted with bias, scale factor and measurement noise, and real flight data that are not corrected for the bias and the scale factor errors, are presented.
- Chapter 6 discusses the conclusions drawn on the basis of results obtained.

Chapter 2

Artificial Neural Networks

2.1 Introduction

"A Neural Network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is based on the biological neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns."

Artificial neural networks (ANNs) are computational paradigms which implement simplified models of their biological counterparts, biological neural networks. Biological neural networks are the local assemblages of neurons and their dendritic connections , called weights. Accordingly, ANNs are characterized by

- Local processing in artificial neurons
- Massively parallel processing, implemented by rich connection pattern between the neurons
- The ability to acquire knowledge via learning from experience
- Knowledge storage in distributed memory

Although the initial intent of ANNs was to explore and reproduce human information processing tasks such as speech, vision, and knowledge processing, ANNs also demonstrated their superior capability for classification and function approximation problems. This has great potential for solving complex problems such as systems control, data compression, optimization problems, pattern recognition, and system identification. One of the application areas to Aerospace engineering problems is in the design of flight control systems based on artificial neural networks. In the present work, we have enhanced the scope of applicability of neural networks for aircraft parameter estimation.

There are many different ways in which we refer to the same type of neural networks technology. Neural networks are described as connectionist systems, because of the connections between individual processing nodes. They are sometimes called adaptive systems, because the values of these connections can change so that the neural network performs more effectively. They are also sometimes called parallel distributed processing systems, which emphasize the way in which the many nodes or neurons in a neural network operate in parallel.

Among all types of ANNs, the FFNNs offer more flexibility, and thereby, have higher potential for future application for modelling aircraft dynamics and aircraft parameter estimation.

2.2 Feed Forward Neural Networks

FFNNs are multi layered perceptrons consisting of an input layer, an output layer and one or more hidden layers. The number of neurons (nodes) in the input and output layers are determined by the number of input and output variables respectively and the number of neurons in the hidden layer is decided by the complexity of the problem.

FFNNs are static in nature and are characterized by unidirectional flow of

variables i.e., inputs are propagated through the hidden layers to the output layer. Each node in the layer (input or hidden) is connected to each node in the next layer (either the hidden or output) through a connective weight. The nodes in the hidden layer(s) and output layer have a nonlinear activation function, e.g., sigmoidal or hyperbolic tangent function. The hidden nodes in FFNN have good approximation capability and thereby allow the network to build a model of arbitrary complexity between input(s) and output(s).

Neural Network can operate only with data. During modelling with neural networks, it is necessary to specify the structure of the pattern because the proper selection and representations of patterns greatly decides the performance of the network. The basic elements of neural network learning phase are called neurons, where all computing is performed. Fig. 2.1 illustrates the most common type of a simulated neuron. Each neuron collects the information from its abutting connections and produces a single output value.

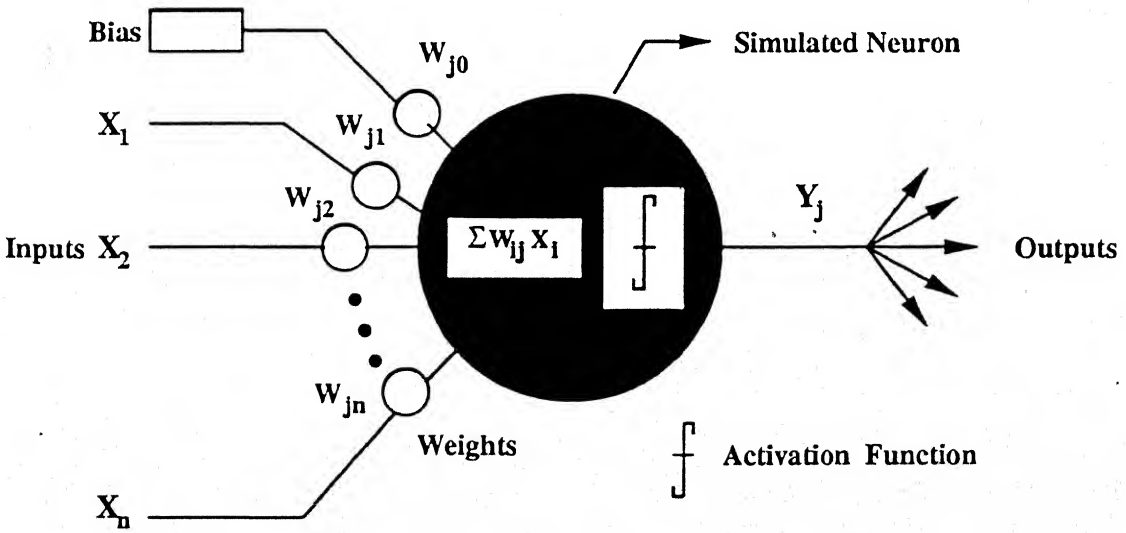


Fig. 2.1 Schematic diagram of a simulated neuron.

2.2.1 Training Phase of FFNN

During the training phase of FFNN, it tries to memorize the pattern of the learning data set. This phase consists of the following modules:

1. Selection of neuron characteristics
2. Selection of Topology
3. Error minimization procedures
4. Stopping Criteria of training

Selection of Neuron characteristics

Neurons can be characterized by two operations, namely, the aggregation and nonlinear filtering. Nonlinear filtering (some times called, Activation Function) can be characterized by several functions, the sigmoidal and the tangent hyperbolic functions being the most common.

Since most of the ANNs use the gradient method to minimize the error, points where the gradients are small should be avoided in the beginning of the process. Therefore to initialize the learning process, the central part of the function where the gradients are large is selected. This can be achieved by scaling proper input signals and the selection of the limit between which random weights are generated. For example, if the scaling of input is carried out between 0.1 and 0.9 then the random weights are preferred between -0.5 and +0.5 for standard sigmoidal function. A major drawback of these activation functions is it's saturation which can cause significant errors when the output has no upper bounds.

Selection of Topology

Topology of FFNN deals with i) number of layers and neurons in each layer and ii) their interconnections. Too few hidden neurons hinder the learning process, while too many occasionally degrade the FFNN generalization capability. Unfortunately, there are no definite rules for selection of the topology of the network and yet, one has to find the best possible topology by trial and error, guided by a set of thumb rules available in literature.

Error minimization procedures

When FFNNs are trained, the weights are modified in order to minimize the error on the training patterns. Different functions can be used to calculate the error. The most common being the MSE and is defined by

$$MSE = \frac{1}{m \times n} \sum_{j=1}^{n} \sum_{i=1}^{m} [Y_i(j) - X_i(j)]^2 \tag{2.1}$$

where Y and X are, respectively, the desired and the predicted outputs; n is the number of data points; and m is the number of output variables.

Different strategies are used for updating the weights of FFNN. In one approach, the weights are updated at every cycle of the entire set of training pattern presentations. This process is called the batch or periodic updating. In the second approach, the networks are updated continuously after each training pattern is presented. This approach is called online or continuous updating.

Training of FFNN is influenced by selection of different tuning parameters. Learning rate (μ) and Momentum rate (Ω) influences the speed of learning in gradient descent technique. Increasing the learning rate may improve the speed of learning but it can also cause oscillations. A proper selection of learning rate parameter helps in achieve fast convergence to accurate estimates.

Even scaling parameters of sigmoidal function (well known as, Logistic gain) also influence the learning behaviour.

Termination Criteria for learning

The processing of adjusting weights are generally repeated until the termination condition is achieved. So in practice, one has to specify termination conditions. The learning process can be terminated when

- the error goes below a specific value
- the magnitude of gradients reached below a certain value
- specific number of iterations (I) is complete

One can also terminate the learning process by cross validation technique. In this technique, data are divided into a training set and validation set. The training set is used to modify the weights, the validation set is used to estimate the generalization ability. Training is stopped when the error on validation set begins to rise. This technique may not be practical when only small amount of data is available. The first three criteria are sensitive to the choice of tuning parameters and may lead to poor results if the tuning parameters are not properly selected.

2.2.2 Training Algorithm - Back Propagation

Back propagation network typically begins with a random set of small weights. The network then adjusts the weights each time it processes an input-output pair. Correction of weight in each pair requires two stages: a forward pass and a backward pass. In forward pass, outcome from input data is generated by flowing the signals in forward direction, from input to hidden and then from hidden to output layer. During the backward pass, the network's actual output

(from forward pass) is compared with the target output and error estimates are computed. These errors are propagated in the backward directions to correct the weights between the two neurons in the subsequent layers.

BPA has been designed to recognize any nonlinear relation between input and output. In the absence of non-linearity, it implements a linear transformation at each layer. The nonlinear function evaluated at each neuron in middle and output layer(s) is called an activation function. Sigmoidal function is generally used as an activation function; it is continuous, monotonically increasing, continuously differentiable and asymptotically approaches fixed finite values as the input approaches infinity (+ or -). An example of Sigmoidal function is

$$f(x) = \frac{1}{1 + e^{(-x/\lambda)}} \tag{2.2}$$

where λ = Logistic gain.
The effect of 'λ' is to modify the shape of sigmoidal function.

Finally, in brief, the steps involved in BPA are:

1. Preprocess or scale the data.
2. Generate the initial random weights.
3. Evaluate the network output for a data set.
4. Calculate the error signal.
5. Modify the weights using the error signal.
6. Repeat from (3) to (5).
7. Repeat from (3) to (6).

Chapter 3

Parameter Estimation Methods

3.1 General

In this chapter, we describe the salient features of the Delta method and the Zero method used for estimating the aircraft parameters from flight data. Both the techniques use the motion variables and control inputs as the input file while the aerodynamic coefficients are presented as the output file for training the neural network. The trained network is a neural model of the aircraft aerodynamics. For the purpose of parameter estimation, the trained neural network is presented with a suitably modified input file (these modifications are explained in later sections), and the corresponding predicted output file of aerodynamic coefficients is obtained. Suitable interpolation and manipulation of such input-output files yields the estimated values of the unknown parameters.

3.2 The Delta method

The stability and control derivatives occurring in the equations of motion of an aircraft represent partial derivatives of aerodynamic force and moment coefficients with respect to the corresponding motion or control input variables.

In other words, these derivatives can be thought of as variations in the aerodynamic coefficient due to small variation in one of the motion or control variables about the nominal value in such a way that only that particular variable is allowed to change, while the rest of the variables are held constant at their nominal values. This fundamental understanding is exploited in the Delta method for estimating parameters.

Let us now assume that FFNN is already trained to map the network input variables to the output variables. Now only one of the network inputs is given a small (delta, Δ) perturbation at each time point (hence the name Delta method) while all others are held at their original values. Such a modified input file is now presented to the trained neural network to predict the perturbed value of aerodynamic coefficients (output) at its output node. The difference in the predicted value of the aerodynamic coefficient from the value predicted for the original values of inputs is due to the perturbation in the value of the chosen network input variable. The difference so calculated in the value of aerodynamic coefficient divided by the perturbation value yields the corresponding aircraft parameter.

For example, say β, is perturbed by small value $\Delta\beta$ and difference observed in the predicted value of C_l for β and $\beta + \Delta\beta$ is ΔC_l. Then $\frac{\Delta C_l}{\Delta\beta}$ yields the stability derivative $C_{l\beta}$. Similarly perturbation in δa and corresponding ΔC_n observed will yield control derivative $C_{n\delta a}$, where $C_{n\delta a} = \frac{\Delta C_n}{\Delta\delta a}$. To avoid bias due to one sided differences, both motion and control variables are perturbed in both increasing and decreasing direction. For example C_y values for $\beta + \Delta\beta$ and $\beta - \Delta\beta$ are predicted to be $C_y{}^+$ and $C_y{}^-$ respectively, then

$$C_{y\beta} = \frac{C_y{}^+ - C_y{}^-}{2\Delta\beta}. \tag{3.1}$$

3.3 The Zero method

The Zero method is based on interpreting the stability and control derivatives as follows: if we could obtain variation in the value of an aerodynamic coefficient due to variation in only one of the motion/control variables while the other motion/control variables are identically zero, then the ratio of the aerodynamic coefficient to the non-zero motion/control variable will yield the corresponding stability/control derivative. Let us say that the FFNN is trained to map the network input variables $(\frac{ps}{V})$, $(\frac{rs}{V})$, β, δa and δr to the network output variable C_l. Now one input variable at a time is chosen to be at its original value while the rest of the network inputs are set to zero (hence the name Zero method). The predicted value of the aerodynamic coefficient corresponding to such a modified file is divided by the non-zero motion/control variable to yield the corresponding stability/control derivative.

To illustrate the method for the lateral-directional case, let us say the rolling moment coefficient has the functional form $C_l = \mathrm{f}(\frac{ps}{V}, \frac{rs}{V}, \beta, \delta a, \delta r)$. Let C_l can be written as

$$C_l = C_{lp}(\frac{ps}{V}) + C_{lr}(\frac{rs}{V}) + C_{l\beta}\beta + C_{l\delta a}\delta a + C_{l\delta r}\delta r \tag{3.2}$$

Although it is not required to postulate an aerodynamic model of Eq. (3.2), it has been done only to facilitate explanation of the Zero method. After the network has been trained to achieve the functional relation, it is used for the prediction phase as follows. The trained network is allowed to predict C_l corresponding to the input file consists of ($\frac{ps}{V}$, $\frac{rs}{V}$=0, β=0, δa=0 and $\delta r = 0$). Let the predicted C_l be denoted by C_l^*. The C_l^* represents the variation in C_l with $(\frac{ps}{V})$ while r=β=δa=δr=0. As evident from Eq. (3.2), the ratio of C_l^* to $(\frac{ps}{V})$ will yield the derivative $C_{lp} = \frac{C_l^*}{(\frac{ps}{V})}$. Similarly all the parameters of the above equation can be estimated by using suitably modified input files. Furthermore, the trained networks using C_n and C_y as the output files can be used to estimate the C_n and C_y derivatives.

The above procedure will be valid when C_{l0}, C_{n0} and C_{y0} are identically zero. If C_{l0}, C_{n0} and C_{y0} have non-zero values, the procedure needs modification as follows: after the aerodynamic coefficients are predicted as before we would need to estimate the trim values of force and moment coefficients (C_l, C_n and C_y) ahead of estimating parameters. For this purpose, all the network inputs are set to zero and the corresponding predicted aerodynamic coefficients would be the desired estimates of trim values. Let the estimated values be C_{l0}, C_{n0} and C_{y0}. Now the parameter estimates are obtained by dividing $(C_l^* - C_{l0})$, $(C_n^* - C_{n0})$ and $(C_y^* - C_{y0})$, instead of just C_l^* , C_n^* and C_y^* as was done earlier, by the corresponding non-zero motion/control variables. For example,

$$C_{lp} = \frac{C_l^* - C_{l0}}{(\frac{ps}{V})}, \quad C_{n\beta} = \frac{C_n^* - C_{n0}}{\beta}, \quad C_{y\delta a} = \frac{C_y^* - C_{y0}}{\delta a}, \quad etc. \qquad (3.3)$$

A few additional considerations common to both the proposed methods are discussed next.

It is noted that both the methods yield one estimated value at each time point, i.e., one estimate for each of the training pair used. Ideally all such values should have been identical to yield a single value of the parameter. However, in reality, there are variations in the predicted values. It is conjectured that such variations may be due to a less-than-perfect match between the actual and the predicted values of the network output variables C_l, C_n and C_y. This suggested that, after the estimates are arranged in a descending or an ascending order, extreme values may be ignored. Practical experience with many sets of simulated data guided the removal of 25% estimates from either end of the so ordered set of estimates. To study the spread of estimates, histograms of all the estimated values were plotted (Fig. 5.5, discussed later). Since the histograms showed a near normal distribution, the mean value is used as the estimated value and the standard deviation about the mean as the measure of accuracy of the estimates.

Chapter 4

Aerodynamic Modelling using Feed Forward Neural Networks

4.1 General

In this chapter, we describe the types of simulated flight data generated and real flight data taken for our study. Details of example airplane used and the type of control inputs utilized for generating simulated flight data are given. Flight data with biases, scale factors, with and without pseudo measurement noise, and with time shift were prepared for analysis. Next, the data pre-processing done for the real flight data is discussed; both the processed data (corrected for biases, scale factors and time shifts) as well as the raw data (only corrected for time shifts) are used for parameter estimation via the Delta and the Zero methods[12]. Finally, the FFNN model for lateral-directional aircraft aerodynamics is presented.

At the outset, it may be emphasized that we need neither the equations of motion nor their solutions for the FFNN modelling and subsequent parameter estimation via the proposed methods. It is only for the purpose of generating simulated flight data that the equations of motion are considered.

4.2 Simulated Flight Data Generation

The example aircraft selected for generating simulated flight data is the research aircraft ATTAS for which we were supplied real flight data by the Institute of Flight Mechanics, DLR, Germany, for analysis at our end. The nominal values of mass and moment of inertia used for data generation, and the geometrical characteristics of the ATTAS aircraft are summarized in Table 4.1.

Table 4.1 Geometric, mass and inertia characteristics of ATTAS aircraft (Ref. 15)

Mean chord	c	=	3.159 m
Half wing span	s	=	21.50 m
Wing area	S	=	64.00 m^2
mass	m	=	16352.23 kg
Moment of inertia	I_x	=	162314.20 kg-m^2
	I_y	=	252687.00 kg-m^2
	I_z	=	388440.00 kg-m^2
	I_{xz}	=	11442.00 kg-m^2

The six-degree-of-freedom equations of motion[15] in a body-fixed axis system fixed to the center of gravity of the aircraft are given by Eq. (4.1).

$$
\begin{aligned}
\dot{u} &= -qw + rv - g\sin\theta + \frac{\bar{q}S}{m}C_X \\
\dot{v} &= -ru + pw + g\cos\theta\sin\phi + \frac{\bar{q}S}{m}C_Y \\
\dot{w} &= -pv + qu + g\cos\theta\cos\phi\frac{\bar{q}S}{m}C_Z \\
\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\
\dot{\psi} &= q\sin\phi\sec\theta - r\cos\phi\sec\theta \\
\dot{p} &= \frac{1}{I_xI_z - I_{xz}^2}\left\{\bar{q}Ss(I_zC_l + I_{xz}C_n) - qr(I_{xz}^2 + I_x^2 - I_yI_z) + pqI_{xz}(I_x - I_y + I_z)\right\} \\
\dot{q} &= \frac{1}{I_y}\left\{\bar{q}Sl_\mu C_m - (p^2 - r^2)I_{xz} + pr(I_z - I_x)\right\} \\
\dot{r} &= \frac{1}{I_xI_z - I_{xz}^2}\left\{\bar{q}Ss(I_xC_n + I_{xz}C_l) - qrI_{xz}(I_x - I_y + I_z) + pq(I_{xz}^2 - I_xI_y + I_x^2)\right\}
\end{aligned}
\tag{4.1}
$$

For the purpose of simulating lateral-directional dynamics, Eq. (4.1) are reduced to three-degree-of-freedom equations of motion as shown below:

$$\begin{aligned}
\dot{v} &= -ru + g\sin\phi + \left(\frac{\overline{q}S}{m}\right) C_y \\
\dot{p} &= \frac{1}{I_x I_z - I_{xz}^2} \{\overline{q}Ss(I_z C_l + I_{xz} C_n)\} \\
\dot{r} &= \frac{1}{I_x I_z - I_{xz}^2} \{\overline{q}Ss(I_x C_n + I_{xz} C_l)\} \\
\dot{\phi} &= p
\end{aligned} \tag{4.2}$$

where,

$V = \sqrt{u^2 + v^2 + w^2}$

$\beta = \frac{v}{V}$

$\overline{q} = \frac{1}{2}\rho V^2$

The aircraft is assumed to be symmetrical about the X-Z plane, and hence, the cross products of inertia I_{xy} and I_{yz} are equal to zero. Also, the thrust components are all neglected. The equations of motion are solved for the flight condition defined by a landing flap deflection δ_f = SP (1 deg), indicated airspeed V_{IAS} = 160 Kn, and using the estimated values of parameters (via ML method) in Ref. 15 as the true values; the true values of parameters used are shown in Table 4.2. The Eq. (4.2) are solved using the fourth-order Runge-Kutta algorithm. The moment and force coefficients (C_l, C_n and C_y) are calculated by using Eq. (4.3)

$$\begin{aligned}
C_l &= C_{l0} + C_{lp}\left(\frac{pb}{2V}\right) + C_{lr}\left(\frac{rb}{2V}\right) + C_{l\beta}\beta + C_{l\delta a}\delta a + C_{l\delta r}\delta r \\
C_n &= C_{n0} + C_{np}\left(\frac{pb}{2V}\right) + C_{nr}\left(\frac{rb}{2V}\right) + C_{n\beta}\beta + C_{n\delta a}\delta a + C_{n\delta r}\delta r \\
C_y &= C_{y0} + C_{yp}\left(\frac{pb}{2V}\right) + C_{yr}\left(\frac{rb}{2V}\right) + C_{y\beta}\beta + C_{y\delta a}\delta a + C_{y\delta r}\delta r
\end{aligned} \tag{4.3}$$

Table 4.2 **True values of parameters used for generating simulated flight data (Ref. 15)**

C_{lp} = -0.9782	C_{np} = -0.1153	C_{yp} = 0.3029
C_{lr} = 0.4181	C_{nr} = -0.4949	C_{yr} = 0.7273
$C_{l\beta}$ = -0.1264	$C_{n\beta}$ = 0.2805	$C_{y\beta}$ = -1.1328
$C_{l\delta a}$ = -0.2469	$C_{n\delta a}$ = 0.0	$C_{y\delta a}$ = 0.0293
$C_{l\delta r}$ = 0.0465	$C_{n\delta r}$ = -0.1659	$C_{y\delta r}$ = 0.1914
C_{l0} = 0.00099	C_{n0} = 0.00161	C_{y0} = -0.00454

4.2.1 Control Input Forms

The multi-step 3-2-1-1 type inputs are capable of better excitation of aircraft dynamics to increase the information content in the responses, and thereby, lead to better estimates via any of the existing parameter estimation methods such as the maximum likelihood method and the filter error method. Similar observations for the Delta and the Zero methods[12] have been confirmed in Refs. 13 and 14. A multistep 3-2-1-1 type of input consists of control deflection for 3,2,1 and 1 second in +ve, -ve, +ve, and -ve direction respectively. The maximum magnitude was kept at 0.1 radians. In the present work, multi-step 3-2-1-1 type aileron and rudder inputs, aileron input followed by, after a 1 second interval, rudder input were used, and corresponding forced response of 8 sec duration was used for training the FFNN. The control inputs used for simulation along with the response obtained are shown in Fig. 4.1, and the computed aerodynamic coefficients are shown in Fig. 4.2.

4.2.2 Different types of simulated flight data used for parameter estimation

The simulated flight response so obtained is analyzed as it is, and also after introducing varying amounts of bias, scale factor, measurement noise, and time shift errors in the motion variables; the value of bias and scale factor is a multiple of the values estimated for the real flight data as given in Ref. 15.

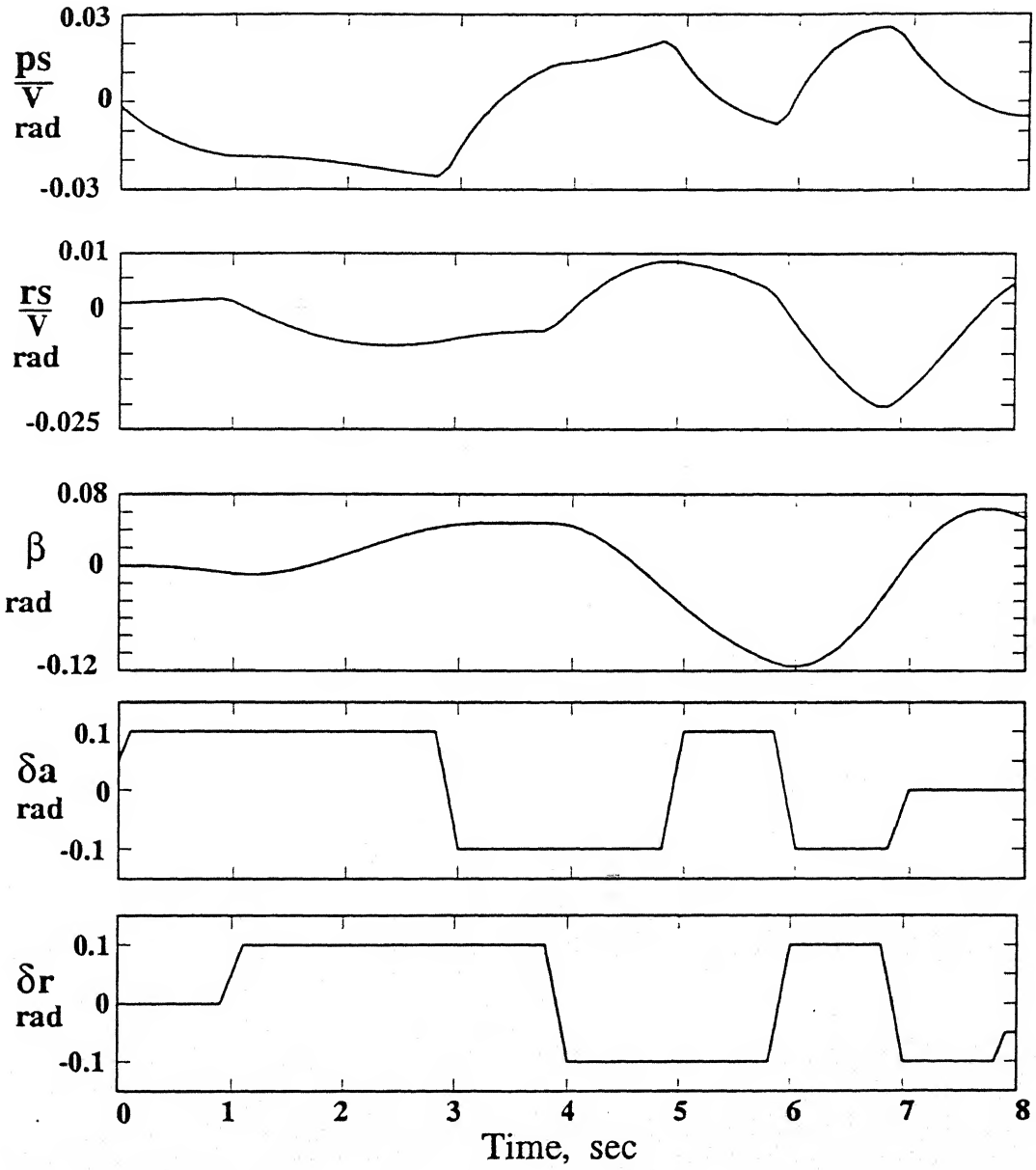


Fig. 4.1 Simulated flight data for a multi-step 3-2-1-1 δa and δr inputs.

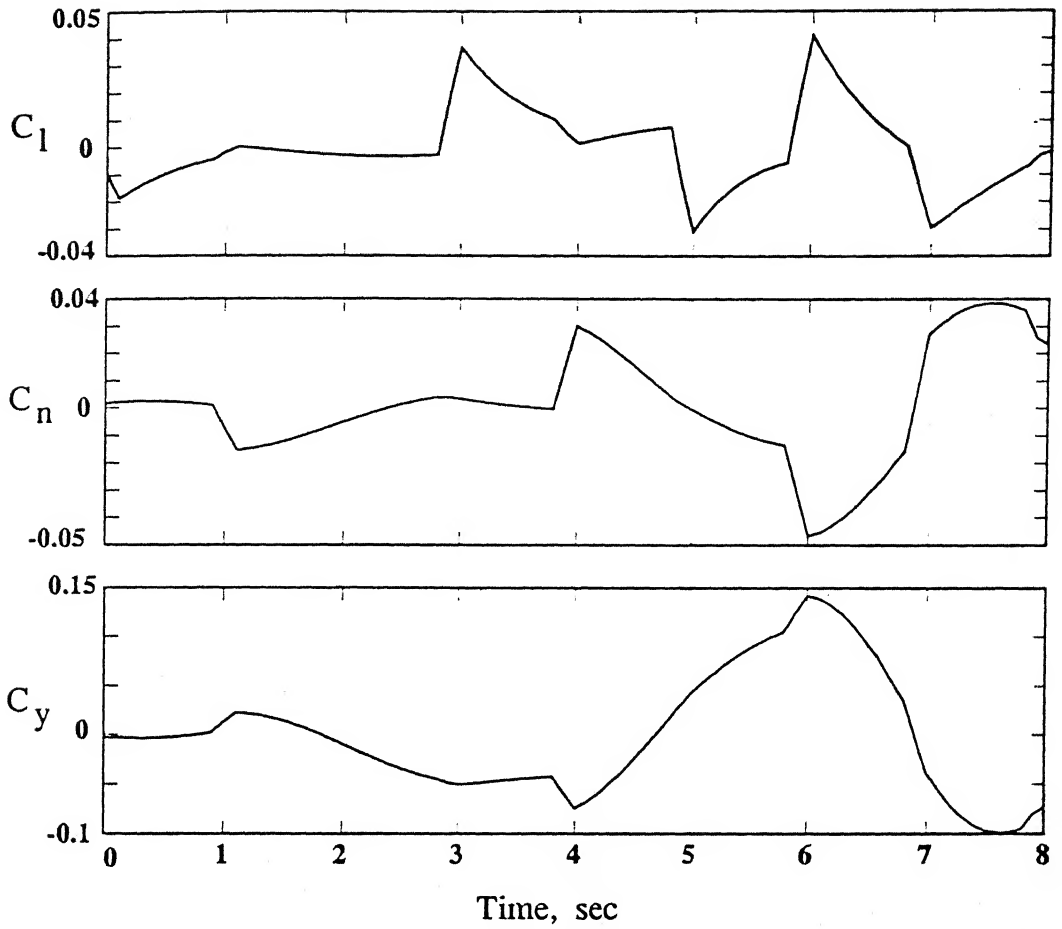


Fig. 4.2 Computed total aerodynamic coefficients for simulated flight data, of Fig. 4.1.

The values of biases and scale factors as in Ref. 15 are shown in Table 4.3.

Table 4.3 The bias, scale factor and time shift estimates of Ref. 15

Parameter	Scale factor	Bias	Time shift (sec)
p rad/sec	-	$p_b = 2.594 \times 10^{-4}$	0.05
q rad/sec	-	$q_b = 2.468 \times 10^{-4}$	0.05
r rad/sec	-	$r_b = -1.574 \times 10^{-4}$	0.05
α rad	1.069	$\alpha_b = 2.14 \times 10^{-3}$	-
β rad	1.075	$\beta_b = -3.767 \times 10^{-3}$	-

To study the effect of noise in simulated flight data, different levels of pseudonoise (1, 5 and 10%) was added to p, r, β, C_l, C_n and C_y. This pseudonoise was simulated by generating successively uncorrelated pseudo-random numbers having a normal distribution with zero mean and assigned a standard deviation, the standard deviation corresponding approximately to the designated percentage (1 and 5%) of the maximum amplitude of the corresponding variable. To study the effect of bias in flight data, varying amounts of biases were added to the p, r, β responses; the biases p_b, r_b and β_b were 1 to 10 times the values shown in Table 4.3. To study the scale factor effects, the β response was corrupted with the scale factor equals to 1.075 (as shown in Table 4.3). Finally, to study the effect of time shift, flight data with time shift values of Table 4.3 were prepared for analysis.

The main findings of the present study are illustrated by presenting results for the following cases of simulated data.

Case 1 : Simulated flight data with no measurement errors.

Case 2 : data of case 1, but specific amount of measurement noise (1%, 5% and 10%) added to p,r,β,C_l,C_n and C_y.

Case 3 : data of case 2, but with p, r, β corrupted with bias equals to the estimated values for the real flight data, i.e., p+p_b, r+r_b, and β+β_b are assumed to be the signals available for training the FFNN.

Case 4 : data of case 3, but β corrupted with scale factor $k_\beta = 1.075$, which equals the value estimated for the real flight data. Thus, the β signal available is assumed to be given by $k_\beta\beta$+β_b.

Case 5 : data of case 2, but with p, r, β corrupted with bias equals to 5 times the estimated values. The signals available are p+$5p_b$, r+$5r_b$, and β+$5\beta_b$.

Case 6 : data of case 5, but β corrupted with scale factor $k_\beta = 1.075$. The β signal becomes $k_\beta\beta$+$5\beta_b$.

The simulated flight data for case 2 to case 6 were analyzed for varying amounts of measurement noise (1%, 5% and 10%). To facilitate discussion of results, the type of data for a specific case (2 to 6) is identified by the following notation. Say, case 3N5 is used to denote case 3 data with noise equals to 5% (N5 stands for noise = 5%). Similarly case 2N1 means case 2 data with 1% noise, and case 6N10 means case 6 data with 10% noise.

4.3 Real Flight-Data Supplied by DLR for ATTAS Aircraft

The real flight data of the ATTAS aircraft supplied (shown in Fig. 4.3) by the Institute of Flight Mechanics, DLR, Germany, consists of raw data for measured u, α, β, p, q, r, a_x, a_y, a_z, $\dot{p}$, $\dot{r}$, H, T, δa, and δr, and information about the fuel consumption and the locations of all the measuring instruments. The measurement of true airspeed (V), angle-of-attack (α) and angle of side-slip (β) were obtained with a flight log mounted on a boom in front of the aircraft nose. The angular rates p,q and r were obtained from measurements available from the inertial platform. The accelerations along the three body fixed axes were measured using an accelerometer triad located near the center of gravity of the aircraft. The angular accelerations $\dot{p}$ and $\dot{r}$ were obtained by numerical differentiation of the corresponding angular rates. The control surface deflections (δa and δr) were measured using potentiometers. The digital air data computer was used to record altitude (H) and outside air temperature (T).

4.3.1 Preprocessing of Real-Flight Data

Real flight-data generally have unknown biases, scale factors and time shifts in the recorded data. These introduce data incompatibility. Thus it is imperative that the data compatibility check is carried out before using the data for aerodynamic modelling and parameter estimation by conventional methods

like the ML method.

Using the estimated biases, scale factors and the known offset-distances of measuring instruments from the center of gravity[15], the measured flight variables that are required for FFNN modelling, are corrected as follows: The corrected value of the side-slip angle β is obtained from

$$\beta = sin^{-1}(\frac{v}{V}) \tag{4.4}$$

where v = $v_n + p\ Z_n + r\ X_n$
and v_n = V sin$((\beta_{nm}$ - $\Delta\beta)/\ k_\beta)$

The β_{nm} is the side-slip angle measured at the nose vane, V is the measured airspeed, k_β and $\Delta\beta$ are the estimated scale factor and bias respectively, X_n and Z_n are the distances of vane from the C.G. along the X and Z directions respectively.

The corrected value of lateral acceleration a_y at C.G of aircraft is obtained from the accelerometer recording along the Y-axis using

$$a_y = a_{ym} - (pq + \dot{r})X_a + (r^2 + p^2)Y_a - (rq - \dot{p})Z_a \tag{4.5}$$

where a_{ym} is the measured value at a point X_a, Y_a and Z_a distances away from the center of gravity.

The measured angular rates (p_m,q_m and r_m) are corrected for the bias error to obtain p, q and r as follows:

$p = p_m$ - p_b; $\quad q = q_m$ - q_b; $\quad r = r_m$ - r_b

The angular rates p, q and r are also corrected for the time delays by shifting the time tag of the measured quantity by the estimated time shift. Estimates of time delays suggested that the recorded values of p, q and r (available at interval of 0.05 sec.) be shifted by one time interval of 0.05 sec., i.e., the nth value be labelled as (n-1)th[15].

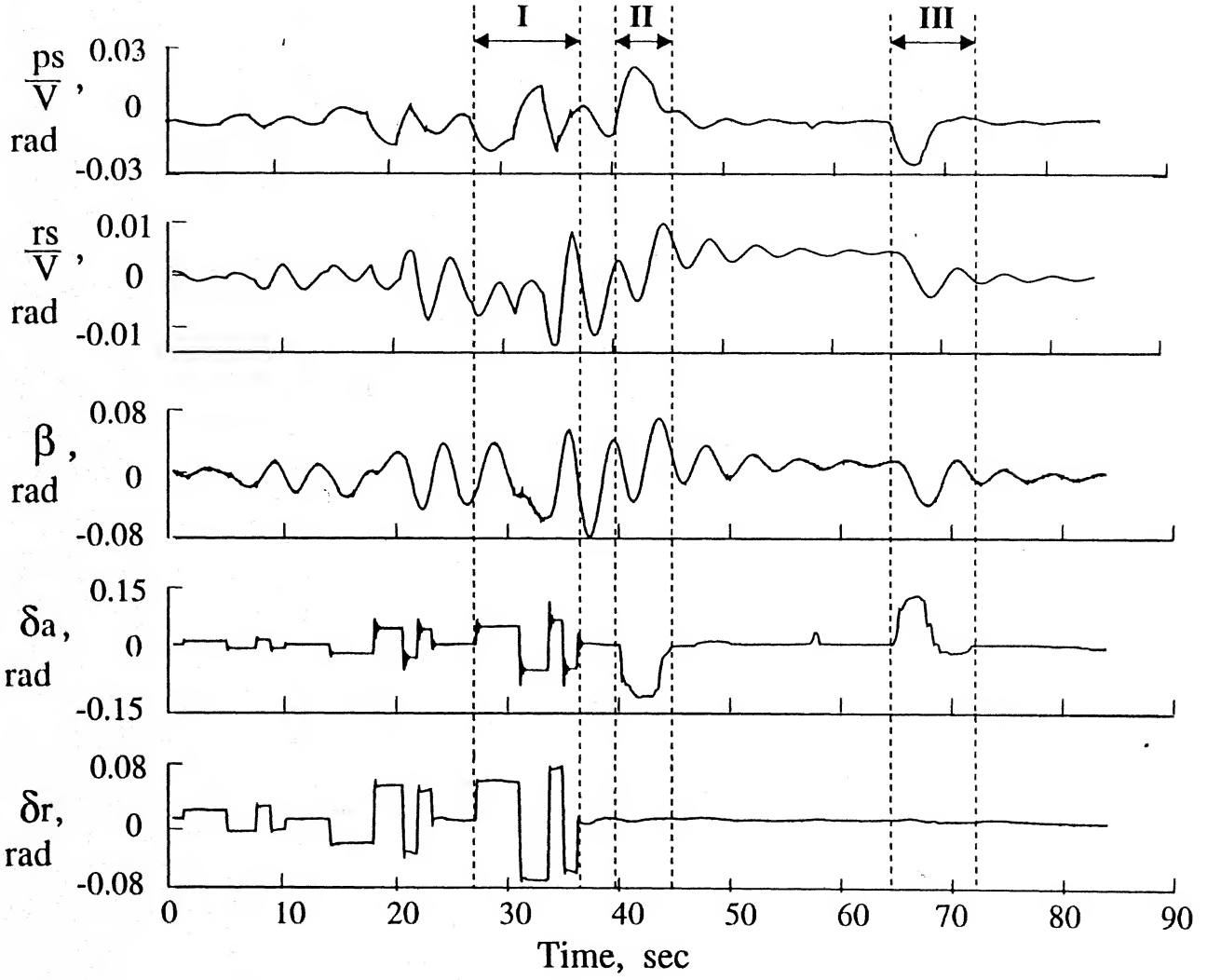


Fig. 4.3 Real flight data of the ATTAS aircraft supplied by DLR.

Thus the raw data supplied by DLR was corrected for the time delays and bias errors in angular rates, the scale factor and bias error for angle of side-slip and the offset of the measuring accelerometer from C.G. of the aircraft for the lateral acceleration. All the correction factors utilized above are listed in Table. 4.3.

In addition to the above corrections done, for good aerodynamic modelling and parameter estimation, one should know about the aircraft mass characteristics, such as mass, moment of inertia, and position of the center of gravity. The information about fuel consumption supplied with the flight-data was used to incorporate the necessary corrections and thereby estimate the aircraft mass, the moment of inertia and the center of gravity location as a function of time.

The fuel-flow meter recordings for the respective flight segments were integrated to get the total fuel consumed from the time of take-off, and its subtraction from the take-off weight yielded the actual weight at any time during the flight. In order to account for the variation of I_x and I_z due to the change in fuel loading, following least square fits have been used.

$$\begin{aligned} I_x &= 104431.0 + 6.31626(W_f) + 0.0021277(W_f)^2 \\ I_z &= 331864.0 + 5.53291(W_f) + 0.0022477(W_f)^2 \end{aligned} \tag{4.6}$$

where I_x and I_z are the moment of inertia of the aircraft along X and Z axes and W_f is the fuel weight available at any given time during the flight. The variation in I_y and I_{xz} due to fuel load changes were estimated to be small, and therefore, constant value of $I_y = 252687.0$ Kg-m^2 and $I_{xz} = 11442.0$ Kg-m^2 has been used in aerodynamic modelling and parameter estimation. The other two cross product of inertia, namely I_{xy} and I_{yz}, are small and hence neglected. Finally, to account for the variation in the location of center of gravity with fuel consumption, the empirical method outlined in Ref. 15 was used. This yielded the C.G. location corresponding to each time point where the flight data is recorded. A fixed reference point (10.12m, 0m, 2.8m) is chosen to serve

as the origin of the body-fixed axes system. A geometrical reference system (Fig. 4.4) has been defined[15] for the transformation of forces and moments about the C.G fixed axes system to the reference point fixed axes system. Next, the offsets of the reference point from the center of gravity at any point were calculated. Let the position of the C.G. relative to the reference point be denoted by ΔX_{CG}, ΔY_{CG} and ΔZ_{CG}. The roll and yaw moment coefficients referred to the reference point (denoted by superscript R) are given by

$$\begin{aligned} C_l^R &= C_l - C_y\frac{\Delta Z_{CG}}{s} + C_z\frac{\Delta Y_{CG}}{s} \\ C_n^R &= C_n - C_x\frac{\Delta Y_{CG}}{s} + C_y\frac{\Delta X_{CG}}{s} \end{aligned} \tag{4.7}$$

where C_l, C_n and C_y are the moment and force coefficients in body-fixed axes system referred to the center of gravity. These C_l, C_n and C_y are in turn computed from the following equations.

$$\begin{aligned} C_l &= \frac{1}{\overline{q}Ss}\left[\dot{p}I_x - \dot{r}I_{xz} - pqI_{xz} + qr(I_z - I_y)\right] \\ C_n &= \frac{1}{\overline{q}Ss}\left[\dot{r}I_z - \dot{p}I_{xz} + qrI_{xz} - pq(I_x - I_y)\right] \\ C_y &= \frac{m}{\overline{q}S}\left[a_{ym} - (pq + \dot{r})X_a + (r^2 + p^2)Y_a - (rq - \dot{p})Z_a\right] \end{aligned} \tag{4.8}$$

We also defined the roll and yaw moment coefficients in the experimental axes system (shown in Fig. 4.5) by applying the following axes transformations:

$$\begin{aligned} C_l{}^e &= C_l{}^R cos\alpha - C_n{}^R sin\alpha \\ C_n{}^e &= C_l{}^R sin\alpha - C_n{}^R cos\alpha \end{aligned} \tag{4.9}$$

where subscript 'e' denotes moment coefficients in the experimental axes system attach to the reference point.

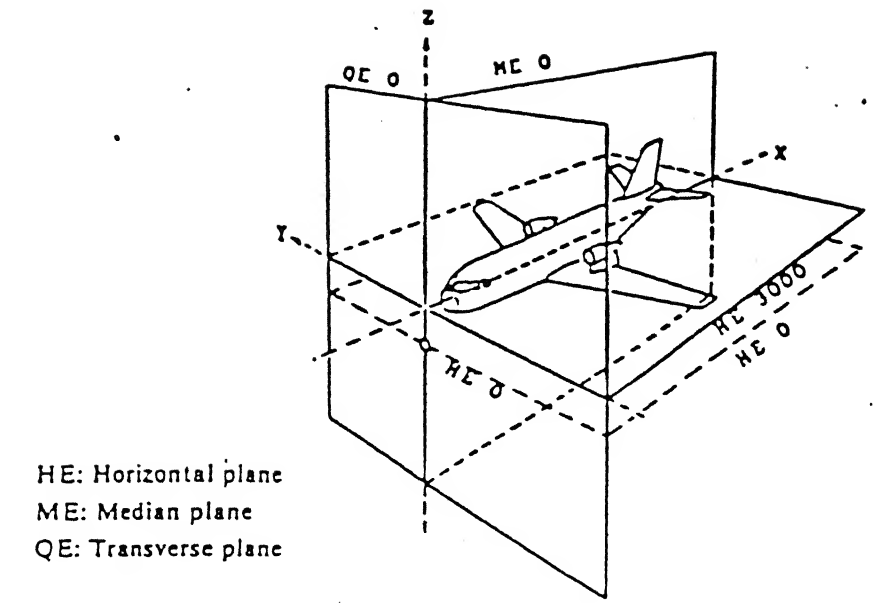


Fig. 4.4 Geometric reference system (Ref. 15).

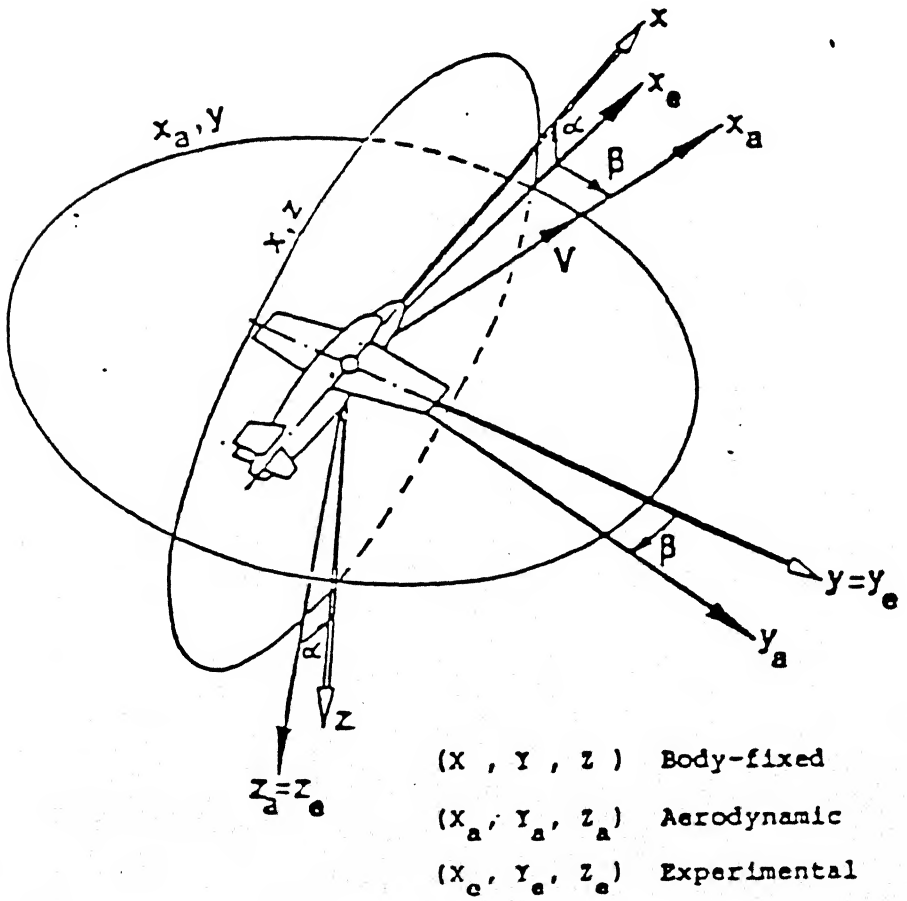


Fig. 4.5 Axes system (Ref. 15)

4.3.2 Two different ways of analyzing the real flight data

The real flight data of ATTAS aircraft (Fig. 4.3) can be used in many different ways. Fig. 4.3 shows three significant segments, denoted by I, II and III, corresponding to where the control inputs are of relatively large magnitude, and thereby, the response is also large. One can estimate the parameters by using the flight segments as follows:

i) considering the whole flight data

ii) considering only the segment I

iii) considering concatenated segments I and II

iv) considering concatenated segments I, II and III

A quick study of all the four options showed that the option (iv) of using concatenated segments I, II and III yielded the best results (the reasons and results are discussed later in Chapter 5). For the data corresponding to the option (iv), the following are the two different ways of analyzing the real flight data, used for parameter estimation via the Delta and the Zero methods.

Case R1 : Recorded (real) flight data corrected for the bias, the scale factor and the time shift errors using their estimates given in Ref. 15 and shown in Table 4.3.

Case R2 : Recorded raw flight data for p, r, β and a_{ym}; p and r corrected for the time shift only.

4.4 Aerodynamic Modelling

The purpose of aerodynamic modelling using FFNNs is to establish a functional relation between the aircraft aerodynamic coefficient and the measurable aircraft motion and control variables.

The choice of the input variables for a particular aerodynamic coefficient is dictated by the physical understanding of the phenomenon governing this dependence. Once the input and output variables for the FFNN are selected, the FFNN model of aircraft aerodynamics is achieved without the need of a formal model structure formulation.

The problem of modelling aircraft aerodynamics by use of FFNN consists of two phases, namely i) learning or training phase and ii) prediction phase.

During the training phase (explained in chapter 2 along with BPA algorithm), the network captures the functional relationship existing between the inputs and the outputs (not necessarily in the same form as in the original physical model) in terms of weights. Once the training is over, the network weights and topology is frozen, and the same input data is passed through the network to check its prediction capability.

A good prediction capability implies good aerodynamic modelling. Thus, a good aerodynamic modelling is imperative for success of the Delta and the Zero methods[12] used for estimating parameters from flight-data. There are many so called influencing or tuning parameters (discussed in chapter 2) of the FFNN that affect the training and thereby prediction capabilities of the network. A proper selection of tuning parameters is necessary for the given set of training data. Unfortunately, there are no definite rules for selection of these network parameters and yet, one has to find the best possible set of parameters by trial and error, guided by a set of thumb rules available in literature. The matrix of tuning parameters that was used to find the best possible architecture is given in Chapter 5.

In the present work, for lateral-directional modelling, the input file consists of roll rate (ps/V), yaw rate (rs/V), angle-of-side-slip (β), aileron deflection (δa) and rudder deflection (δr). The output file has rolling moment (C_l) or yawing moment (C_n) or side force coefficient (C_y). The schematic of FFNN for lateral-directional aerodynamic modelling is shown in Fig. 4.6.

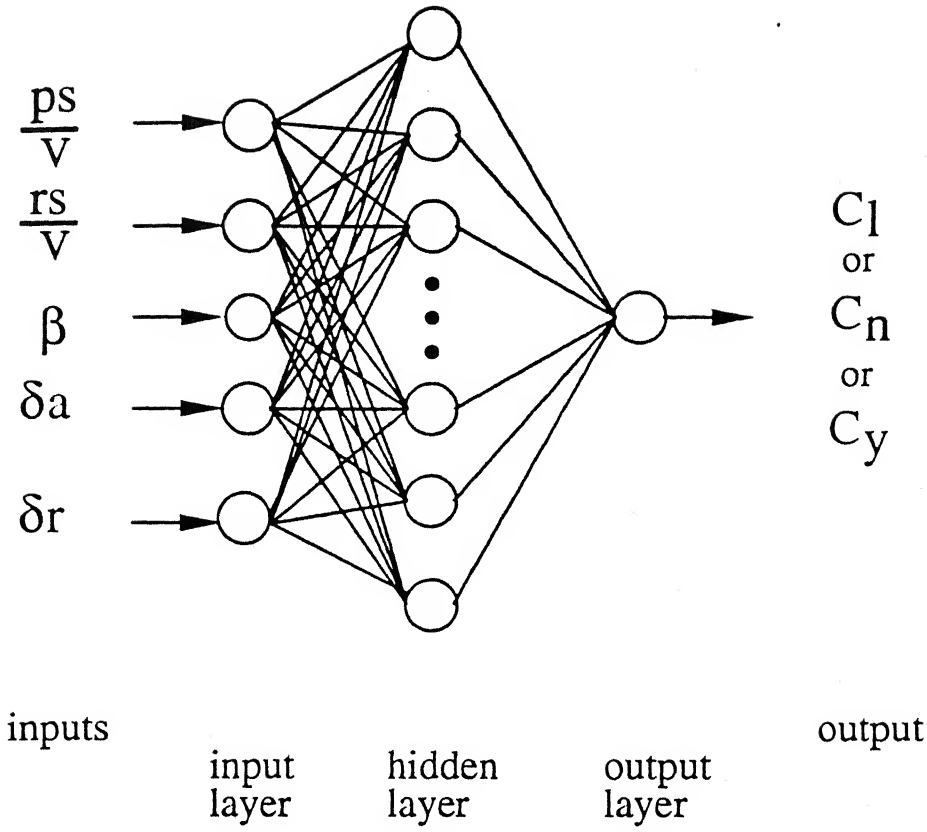


Fig. 4.6 Schematic of FFNN for lateral-directional aerodynamic modelling

It may be clarified that the input variables to the FFNN need not be limited to only those mentioned above. For the simulated flight data, the choice of the input variables is obvious since the variables involved in the equations of motion used for generating the data are known. In the case of real flight data, it may be desirable to include all possible variables of which the aerodynamic coefficients may be a function of. Thus any nonlinear functional relation between the aerodynamic coefficients and motion/control variables can be easily mapped to obtain the aerodynamic model of the aircraft, without any need for an a priori model.

Chapter 5

Results and Discussion

5.1 General

The simulated flight data for lateral-directional dynamics of an aircraft are analyzed for varying amounts of measurement errors in the motion variables. The real flight data with and without correcting for measurement errors are also analyzed. The parameters estimated from both simulated as well as real flight data are presented and compared with the ML estimates of Ref. 15. The network input file consisted of motion variables (p, r, β) along with aircraft control inputs (δa and δr). The network output file is either C_l or C_n or C_y. The network is trained using such input-output files for many such sets of simulated and real flight data.

For each data set, an individual optimal architecture was searched by varying the network tuning parameters; specifically, the variations in the number of neurons in the hidden layers (3-6), the learning rate (0.2-0.8), the momentum rate (0.3-0.9), the random seed (0.3-0.9), the logistic gain (0.7-0.95), and the number of iterations (500-15,000) were tried, and the architecture with the lowest mean square error (MSE) and high correlation coefficient was selected.

Although, we found the best possible optimum architecture, it is possible

that a slight variation in the network tuning parameters may result in still lower MSE for the specific set of flight data. Since the present work is concerned with parameter estimation via the Delta and the Zero methods,[12] we refrain from reporting detailed discussion on training phase and effects of network parameters on the MSE, and thereby on the correlation obtained between the desired and the predicted values of C_l or C_n or C_y. The effect of network tuning parameters on training was similar to that reported in the literature[11,16].

As already described (Chapter 3), the application of the Delta method requires small (Δ) perturbation of network inputs. A brief study was carried out for 0.01 and 0.001 as the perturbation value, and the results were almost identical (marginally better for 0.001). For convenience and consistency, it was, therefore, decided to keep the perturbation fixed at 0.001 for all cases reported in the thesis.

It is worth mentioning that in real life, the $\dot{p}$ and $\dot{r}$ required for computing C_l and C_n would generally be obtained by numerical differentiation of the recorded p and r signals. In this context, for simulated data of case 2 to 6, the only signals of p and r assumed to be available for differentiation are the p and r signals contaminated with the measurement errors. Also for real flight data (case R2), the measured (sensor) signals of p and r are assumed to be available for differentiation, i.e., the recorded p and r signals are used without correcting them for the bias and scale factor errors. This is done to account for any additional inaccuracies that might creep in the parameter estimates through the process of numerical differentiation of the signals having measurement errors.

For the purpose of numerical differentiation, a centrally pivoted 5-point algorithm is used. The algorithm was first checked for signals having no measurement errors in them. A detailed study showed that the computed values of the first derivative via the numerical differentiation compared well with the analytical values except at a few points near the maxima and minima of p and r signals. Practical considerations suggested that a few data points (up to 4)

in the neighborhood of the maxima and minima of the p and r signals be left out of the data set to be used for training the FFNN. Similarly for the real flight data (case R2) also, a few points are omitted; the omitted data are from the neighborhood of the maxima and minima of the raw signals of p and r (containing the bias and scale factor errors). The $\dot{p}$ and $\dot{r}$ so obtained are used to compute the required C_l and C_n from Eq. (4.8).

5.2 Results for Simulated Flight Data

As a first step, a study was undertaken to asses the effect of number of iterations on the accuracy of the estimated parameters. For this purpose, for case 1, the architecture of the network was kept fixed while the number of iterations were varied from 500 to 15,000. It was found that the MSE decreased upto about 5000 iterations, and beyond it, the decrease was only marginal. However, the sample standard deviations and the accuracy of the estimated parameters did improve beyond 5000 iterations, and up to about 10,000 iterations. This was specially true for the weak derivatives as may be seen from Table 5.1, wherein the parameter estimates from simulated flight data via the Delta method and the Zero method and compared for the number of iterations equal to 5000 and 10,000. Thus, the number of iterations was kept fixed at 10,000 for all the cases studied and reported here in.

It is of interest to compare the actual C_l, C_n and C_y with those computed using the estimated values of the parameters. Using the estimated parameters, estimated aerodynamic coefficients C_l, C_n and C_y are obtained from the expression on the right hand side of the Eq. (4.3). Along with these, trained C_l, C_n and C_y from the neural network are also shown in Fig. 5.1. From Fig. 5.1, it may be seen that the actual and trained C_l, C_n and C_y are indistinguishable on the scale of the figure, and the estimated C_l, C_n and C_y corresponding to the Delta method show good match with the actual values.

A detailed study has been carried out to observe the effects all measure-

ment errors in various possible combinations, i.e., the measurement errors are introduced in the data individually, or two at a time, or three at a time, or all at a time. In other words, the simulated data with no measurement errors (case 1) is corrupted with different levels of noise, bias, scale factor and time shift errors. In all the cases studied, it was observed that presence of time shift in any of the motion variables resulted in poor estimates, irrespective of whether any other measurement errors were present or not. Thus, it was concluded that the estimation of parameters from the Delta and the Zero methods required that the data be first corrected for any time shift errors before its use for the training of network, and subsequently for parameter estimation via the Delta or the Zero method. It is, therefore, necessary that the time delay in the recorded data should be estimated either via data compatibility check, or from estimates of possible transport delays in specifications for inertial platform.[15]

The estimated parameters from flight data with varying amounts of measurement errors (measurement noise, bias and scale factor) are presented in the following tables:

Table 5.2 : estimates via the Delta method from simulated data with 1% noise, and bias and scale factor corresponding to case 2 to case 6, i.e., for the cases designated by case 2N1, case 3N1, case 4N1, case 5N1, and case 6N1.

Table 5.3 : estimates via the Zero method for cases same as for Table 5.2.

Table 5.4 : similar to cases for Table 5.2, but with measurement noise increased to 5%, i.e., for cases designated by case 2N5 to case 6N5.

Table 5.5 : estimated via the Zero method for cases same as for Table 5.4.

Table 5.6 : similar to cases for Table 5.2, but with measurement noise increased to 10%, i.e., for cases 2N10 to 6N10.

Table 5.7 : estimated via the Zero method for cases same as for Table 5.6.

Effect of measurement noise on parameter estimates

The estimated parameters for case 2N1 show that in spite of addition of 1% measurement noise to flight data, all the parameters are still well estimated. Comparison of estimates for case 2N1 via the Delta method (Table 5.2) and the Zero method (Table 5.3) shows that the Delta method yields relatively better estimates. This superiority becomes more obvious for higher levels of measurement noise as may be seen by comparing results for the Delta method and the Zero method for case 2N5 and case 2N10 corresponding to 5% and 10% measurement noise (Tables 5.4 to 5.7).

Effect of bias on parameter estimates

The estimated parameters via the Delta method for case 3N1 and case 5N1 in Table 5.2 show that the estimates are hardly affected by the presence of bias in the flight data even in presence of measurement noise increased to 5% (case 3N5, Table 5.4) and 10% (case 3N10, Table 5.6), the parameters are well estimated for biases of (p_b, r_b, β_b) and ($5p_b$, $5r_b$, $5\beta_b$) in flight data. Thus, the Delta method is quite robust with respect to presence of any bias errors in the data.

In contrast, the parameter estimates via the Zero method from the biased data show rapid degradation as the bias and/or measurement noise levels increase. This is clearly by results for case 3 and case 5 for all the three measurement noise levels, i.e., for case 3N1 and case 5N1 in Table 5.3, case 3N5 and case 5N5 in Table 5.5, and case 3N10 and case 5N10 in Table 5.7. It is worth noting that the p and r derivatives like C_{lp}, C_{np}, C_{yp}, C_{lr}, C_{nr} and C_{yr} are poorly estimated for the biased data.

It may be recalled that the Zero method utilizes the predicted total force/moment coefficient for, one at a time, non-zero motion/control variable in the input file, and the stability/control derivative is then computed by

$C_{xi} = \frac{C_x^* - C_{x0}}{i}$, where x = l, m, n; i = p, r, β, δa, δr. The computation involves division by the motion variable i = p, r, or β which has bias as well as measurement noise in it, and also it uses estimated C_{x0} value. It is conjectured that the results via the Zero method are adversely affected due to division by biased motion variables and by inaccurate estimates of C_{x0}.

On the other hand, the Delta method utilizes the generalization properties of the FFNN that allow the network to interpolate and extrapolate, as required. Thus, even if the network is trained on biased and/or noisy data, the prediction of output corresponding to the perturbed, one at a time, network input variable is well performed, and therefore, the estimated parameters given by $C_{xi} = \frac{C_x^+ - C_x^-}{2\Delta i}$ are accurately predicted.

Effect of scale factor on parameter estimates

For parameter estimation via the Delta method, the addition of scale factor to β response seems to affect the β derivatives like $C_{l\beta}$, $C_{n\beta}$ and $C_{y\beta}$, albeit marginally, as shown in Tables 5.2, 5.4 and 5.6 for, respectively, (case 4N1 and case 6N1), (case 4N5 and case 6N5) and (case 4N10 and case 6N10). However, the results for the same cases via the Zero method shown in Tables 5.3, 5.5 and 5.7 indicate that the estimated parameters are poorly estimated. Of course, it must be pointed out that the case 4 and case 6 are for the flight data having noise and bias in addition to the scale factor error. In a separate study (results not given), it was observed that the scale alone did result in good estimates via the Delta method, but most of the estimates via the Zero method were poor.

It is also of interest to compare the actual C_l, C_n and C_y with those computed using the estimated values of parameters (via the Delta method) for case 6N5, where all the measurement errors (except the time shift) are present in the data. Such comparison was shown in Fig. 5.2. Along with these, trained C_l, C_n and C_y from neural network are also shown in the figure. From Fig.

5.2, it may be seen that the trained and the estimated C_l, C_n and C_y are close to each other on the scale of the figure.

To bring out the superiority of the parameter estimates via the Delta method as compared to the Zero method, the parameter estimates from the same flight data (case 6N5) via the both the methods are used to compute the corresponding total coefficients C_l, C_n and C_y using Eq. (4.3). These coefficients are compared in Fig. 5.3 with the actual total coefficients used in generating the simulated flight data for case 6N5. As seen from Fig. 5.3, the coefficients C_l, C_n and C_y based on the Delta method show much better matching with the actual coefficients.

The above results for parameter estimation for simulated data have clearly indicated that the Delta method is quite robust with respect to measurement errors. Thus, for any practical applications wherein the measured data is likely to have measurement errors, the Delta method should be used for parameter estimation.

Effect of utilizing free response for parameter estimation

All the results presented so far, utilized only the forced response of 8 seconds duration corresponding to the 8 seconds duration for which the control inputs δa and/or δr are applied. It is interest to see how the accuracy of estimates is affected by increasing the signal length by including part of the free response that follows the forced response. To this purpose the 8 seconds forced response and the next 4 seconds free response were used together to analyze the resulting 12 seconds response shown in Fig. 5.4 (designated as segment I in Fig. 5.4). The segment I data was analyzed as it is (without introducing any measurement errors) and corresponding to case 6N5 (5 times bias, scale factor and 5% measurement noise) analyzed earlier; the results via the Delta method from segment I are compared with the earlier case 1 and case 6N5 in Table 5.8. For case 1 the results improve marginally when analyzed with free response.

However, for case 6N5, the estimates without free response are slightly better. Again, as discussed for earlier cases, the Delta method yields better results than the Zero method (results not presented for the Zero method). Thus, there seems no need and no advantage to be gained by including any portion of the free response with the forced response when analyzing recorded data for parameter estimation via the Delta method.

Finally, we also investigated if reasonable parameter estimates could be obtained via the Delta and the Zero methods by using only the free response portion of the flight data (segment II in Fig. 5.4). In this case, the input file for training the FFNN consists of the motion variables p, r and β only. The results are shown in Table 5.9. Because the p, r, β responses damp out quickly, the information content in the free response is not adequate, and therefore, the accuracy of the estimates is poor. Therefore, we recommend the parameter estimation should not be attempted using the free response alone.

Before concluding discussion about results for simulated flight data, it is recalled that both methods yield as many estimates of the parameters as there are the data points, since each training pair of the data set results in a corresponding parameter estimate. The spread of these estimates is illustrated in Fig. 5.5 for the estimates obtained via the Delta method for case 6N5. The figure also lists the mean values and the sample standard deviations of the estimated parameters along with the true values for ready comparison. A near-normal distribution of the estimated values is observed for most of the parameters.

5.3 Results for Real Flight Data

The real flight data of case R1 is essentially the same for which parameter estimates via the Delta method are given by Ghosh et al[13] (it is referred to as case 9 in Ref. 13). However, in the present study, the flight data was divided into smaller segments and three segments, marked I, II and III in Fig. 4.3, are

concatenated to yield the flight data for case R1 and case R2. For case R1, the data is corrected for any bias, scale factor and time shift; the estimates of bias, scale factor and time shift from Table 4.3 are utilized for the purpose. For case R2, the data is corrected for the time shift only.

It may be mentioned that the initial conditions for the motion variables were neither required nor estimated; during the network training, the input vector only needs to be paired with the corresponding target vector representing the desired output. This fact can be used advantageously to increase the information content of the input-output pairs selected for the parameter estimation via the Delta and Zero methods[12]: from the total recorded signal, the data corresponding to where the magnitude of the motion variables is relatively large are selected and used for training the network. In case of the Delta and Zero methods,[12] the partitioning of the longer duration manoeuvre into several time segments and their concatenation does not require estimation of unknown initial conditions for each segment separately. On the other hand, the output error methods do necessarily require estimation of initial conditions whenever the data of several segments is concatenated. Thus the Delta and Zero methods[12] can take advantage of concatenated data without the laborious and time consuming requirement of estimating many initial conditions.

Incidentally, it may be mentioned that such partitioning of real flight data was not resorted to in Ref. 13; the present study shows that, from the same real flight data, better estimates are obtained by such concatenation of selected time segments as compared to the results in Ref. 13 obtained by using the total signal length. The estimated parameters for case R1 (for both the Delta and Zero methods) are listed in Table 5.10 along with the estimates from the same data via the Delta method in Ref. 13 and via the ML method in Ref. 15. Assuming the ML estimates of Ref. 15 to be the nominal values for the purpose of comparison, Table 5.10 shows that the use of concatenated data (case R1, the Delta method) results in improved accuracy of estimates as compared to the estimates reported in Ref. 13. Table 5.10 also shows that

the Zero method results are relatively poor as compared to those via the Delta method. The control derivatives $C_{y\delta a}$ and $C_{y\delta r}$ are estimated poorly by both the Delta and the Zero methods; a possible reason for it is discussed later.

Fig. 5.6 shows the comparison of the actual total coefficients C_l, C_n and C_y (obtained by Eq. (4.8), using real flight data), with the estimated coefficients, computed by Eq. (4.3) using the estimated parameters via the Delta method and Ref. 13. The estimated coefficients corresponding to estimates via the Delta method show a better match with actual coefficients than those corresponding to Ref. 13.

Data for case R2 consists of the recorded raw data of p, r, β, and a_{ym}. The p and r signals are corrected for the time shift only as per estimates given in Table 4.3. Thus, the recorded signals are neither checked nor corrected for the presence of measurement noise or the measurement errors like the bias and the scale factor. The raw signals of p and r (after correcting for time shift) are numerically differentiated to get $\dot{p}$ and $\dot{r}$, and these, along with recorded a_{ym} are used to compute C_l, C_n and C_y by Eq. (4.8). The network is trained with the so computed C_l, C_n or C_y as the network output, and the recorded motion and control variables (corrected for time shift only) as the network inputs. The estimated parameters (from both the Delta and the Zero methods) for case R2 are given in Table 5.10; as seen from the table, most of the parameter estimates via the Delta method compare well with the estimates of Ref. 15 and case R1. Furthermore, the parameters via the Zero method are again relatively poor as compared to those from the Delta method. Thus, the earlier observations about the Delta method being more robust with respect to the measurement errors is also confirmed by the results for the real flight data.

A comment about the real flight data analyzed is in order. Although the flight data supplied by DLR are of very high quality, the data have been generated by similar looking δa and δr inputs applied simultaneously (segment I in Fig 4.3) followed by pulse type δa inputs (segments II and III in Fig. 4.3). As pointed out Ref. 13, segment I type of control inputs for generating

flight data are not what one would recommend for generating flight data for parameter estimation via the Delta and Zero methods;[12] the control inputs of δa and δr with a time delay between application of δa and δr, like those used for simulated data (cases 1-6), are to be preferred and would yield better results. The presence of segment I type of control inputs may also be the reason for consistently poor estimation of $C_{y\delta a}$ and $C_{y\delta r}$ for both cases R1 and R2 data analyzed by the Delta and the Zero methods. Notwithstanding the above shortcoming of the real flight data, the applicability of the Delta and Zero methods[12] to raw sensor data for parameter estimation has been demonstrated.

It is of interest to compare the total coefficients C_l, C_n and C_y computed using the parameter estimates obtained via the Delta method for case R2, and the ML method (Ref. 15). Such a comparison shown in Fig. 5.7, along with the actual total coefficients obtained from Eq. (4.8) using the real flight data. To help in distinguishing the three C_l curves, Fig. 5.7 shows a small portion of the C_l versus time plot on a enlarged scale (x4). It may be seen from Fig. 5.7 that the C_l, C_n and C_y based on estimates from neither the Delta method nor the ML method (Ref. 15) show a clear cut or uniformly better matching with the actual total coefficients. Thus, it is difficult to claim superiority of either of the methods, i.e., the Delta method or the maximum likelihood method. On the other hand, it may be pointed out that, unlike the ML method, the Delta method[12] does not require a guess for a reasonable set of initial values of the parameters. Furthermore, the estimates from the proposed methods are obtained by using the recorded data without correcting it for bias or scale factor errors as was required and done before the same data was used for the ML method in Ref. 15.

Table 5.1 Effect of the number of iterations on estimated parameters from flight data of case 1

Parameter	True Value	Delta method I = 5000	Zero method I = 5000	Delta method I = 10000	Zero mehtod I = 10000
MSE	-	650×10^{-9}	646×10^{-9}	556×10^{-9}	538×10^{-9}
C_{lp}	-0.9782	-1.074	-1.197	-1.063	-1.182
		(0.052)†	(0.069)	(0.051)	(0.050)
C_{lr}	0.4181	0.450	0.582	0.430	0.531
		(0.052)	(0.155)	(0.069)	(0.020)
$C_{l\beta}$	-0.1264	-0.128	-0.138	-0.125	-0.136
		(0.009)	(0.011)	(0.008)	(0.005)
$C_{l\delta a}$	-0.2469	-0.251	-0.240	-0.251	-0.240
		(0.012)	(0.022)	(0.014)	(0.025)
$C_{l\delta r}$	0.0465	0.042	0.044	0.042	0.043
		(0.005)	(0.006)	(0.004)	(0.002)
C_{l0}	0.0009	-	-0.0006	-	-0.0005
MSE	-	201×10^{-9}	193×10^{-9}	144×10^{-9}	141×10^{-9}
C_{np}	-0.1153	-0.133	-0.124	-0.108	-0.122
		(0.027)	(0.011)	(0.025)	(0.013)
C_{nr}	-0.4949	-0.589	-0.708	-0.622	-0.723
		(0.141)	(0.018)	(0.058)	(0.017)
$C_{n\beta}$	0.2805	0.279	0.300	0.277	0.294
		(0.014)	(0.032)	(0.013)	(0.008)
$C_{n\delta a}$	0.00	0.007	0.004	-0.002	0.001
		(0.021)	(0.017)	(0.006)	(0.005)
$C_{n\delta r}$	-0.1659	-0.173	-0.181	-0.173	-0.179
		(0.013)	(0.010)	(0.009)	(0.011)
C_{n0}	0.0016	-	0.0022	-	0.0019
MSE	-	697×10^{-9}	665×10^{-9}	443×10^{-9}	429×10^{-9}
C_{yp}	0.3029	0.303	0.307	0.291	0.282
		(0.138)	(0.098)	(0.144)	(0.054)
C_{yr}	0.7273	0.601	0.698	0.667	0.742
		(0.079)	(0.107)	(0.078)	(0.020)
$C_{y\beta}$	-1.1328	-1.095	-1.178	-1.098	-1.212
		(0.082)	(0.075)	(0.075)	(0.048)
$C_{y\delta a}$	0.0293	0.031	0.035	0.025	0.029
		(0.019)	(0.017)	(0.015)	(0.010)
$C_{y\delta r}$	0.1914	0.191	0.198	0.194	0.207
		(0.024)	(0.039)	(0.021)	(0.032)
C_{y0}	-0.0045	-	-0.0059	-	-0.0046

† Sample standard deviation

Table 5.2 Effect of bias and scale factor on parameter estimates via the Delta method. Measurement noise: 1%.

Parameter	True Value	Case 2N1	Case 3N1	Case 4N1	Case 5N1	Case 6N1
C_{lp}	-0.9782	-1.054	-1.054	-1.054	-1.053	-1.053
		(0.054)†	(0.054)	(0.054)	(0.055)	(0.055)
C_{lr}	0.4181	0.431	0.431	0.431	0.431	0.431
		(0.071)	(0.071)	(0.071)	(0.072)	(0.072)
$C_{l\beta}$	-0.1264	-0.124	-0.124	-0.115	-0.123	-0.116
		(0.008)	(0.008)	(0.007)	(0.008)	(0.007)
$C_{l\delta a}$	-0.2469	-0.249	-0.249	-0.249	-0.249	-0.249
		(0.016)	(0.016)	(0.016)	(0.016)	(0.016)
$C_{l\delta r}$	0.0465	0.043	0.043	0.043	0.043	0.043
		(0.004)	(0.004)	(0.004)	(0.004)	(0.004)
C_{np}	-0.1153	-0.116	-0.117	-0.117	-0.118	-0.118
		(0.026)	(0.026)	(0.026)	(0.027)	(0.027)
C_{nr}	-0.4949	-0.600	-0.600	-0.600	-0.600	-0.600
		(0.058)	(0.058)	(0.058)	(0.058)	(0.058)
$C_{n\beta}$	0.2805	0.278	0.278	0.260	0.278	0.260
		(0.012)	(0.012)	(0.012)	(0.013)	(0.012)
$C_{n\delta a}$	0.00	-0.002	-0.002	-0.002	-0.002	-0.002
		(0.006)	(0.006)	(0.006)	(0.007)	(0.007)
$C_{n\delta r}$	-0.1659	-0.172	-0.173	-0.173	-0.172	-0.172
		(0.009)	(0.009)	(0.009)	(0.010)	(0.010)
C_{yp}	0.3029	0.311	0.312	0.313	0.317	0.317
		(0.148)	(0.149)	(0.149)	(0.151)	(0.150)
C_{yr}	0.7273	0.652	0.651	0.651	0.648	0.649
		(0.071)	(0.072)	(0.072)	(0.073)	(0.073)
$C_{y\beta}$	-1.1328	-1.104	-1.104	-1.032	-1.105	-1.032
		(0.074)	(0.074)	(0.069)	(0.074)	(0.069)
$C_{y\delta a}$	0.0293	0.024	0.025	0.025	0.025	0.025
		(0.016)	(0.016)	(0.017)	(0.017)	(0.017)
$C_{y\delta r}$	0.1914	0.194	0.194	0.194	0.194	0.194
		(0.020)	(0.021)	(0.021)	(0.020)	(0.020)

† Sample standard deviation

Table 5.3 Effect of bias and scale factor on parameter estimates via the Zero method. Measurement noise: 1%.

Parameter	True Value	Case 2N1	Case 3N1	Case 4N1	Case 5N1	Case 6N1
C_{lp}	-0.9782	-1.179	-1.188	-1.188	-1.211	-1.209
		(0.052)†	(0.048)	(0.048)	(0.042)	(0.042)
C_{lr}	0.4181	0.537	0.547	0.547	0.583	0.581
		(0.021)	(0.021)	(0.021)	(0.021)	(0.020)
$C_{l\beta}$	-0.1264	-0.133	-0.135	-0.126	-0.139	-0.130
		(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
$C_{l\delta a}$	-0.2469	-0.238	-0.241	-0.241	-0.251	-0.251
		(0.025)	(0.023)	(0.024)	(0.016)	(0.016)
$C_{l\delta r}$	0.0465	0.043	0.043	0.043	0.044	0.044
		(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
C_{l0}	0.0009	-0.0005	0.0001	0.0001	0.0030	0.0028
C_{np}	-0.1153	-0.126	-0.115	-0.116	-0.074	-0.077
		(0.014)	(0.014)	(0.014)	(0.015)	(0.015)
C_{nr}	-0.4949	-0.705	-0.702	-0.703	-0.682	-0.684
		(0.017)	(0.017)	(0.017)	(0.017)	(0.017)
$C_{n\beta}$	0.2805	0.297	0.298	0.279	0.301	0.281
		(0.008)	(0.008)	(0.008)	(0.008)	(0.008)
$C_{n\delta a}$	0.00	0.001	0.003	0.003	0.012	0.011
		(0.007)	(0.008)	(0.008)	(0.010)	(0.010)
$C_{n\delta r}$	-0.1659	-0.179	-0.179	-0.179	-0.174	-0.174
		(0.011)	(0.012)	(0.012)	(0.016)	(0.016)
C_{n0}	0.0016	0.0022	0.0007	0.0008	-0.0052	-0.0047
C_{yp}	0.3029	0.275	0.219	0.223	-0.008	0.010
		(0.057)	(0.054)	(0.054)	(0.038)	(0.039)
C_{yr}	0.7273	0.725	0.711	0.711	0.640	0.647
		(0.019)	(0.019)	(0.019)	(0.017)	(0.017)
$C_{y\beta}$	-1.1328	-1.222	-1.228	-1.148	-1.248	-1.165
		(0.049)	(0.049)	(0.046)	(0.045)	(0.043)
$C_{y\delta a}$	0.0293	0.029	0.022	0.023	-0.009	-0.006
		(0.014)	(0.015)	(0.015)	(0.020)	(0.020)
$C_{y\delta r}$	0.1914	0.209	0.207	0.207	0.194	0.196
		(0.031)	(0.031)	(0.031)	(0.029)	(0.029)
C_{y0}	-0.0045	-0.0047	0.0014	0.0010	0.0267	0.0246

† Sample standard deviation

Table 5.4 Effect of bias and scale factor on parameter estimates via the Delta method. Measurement noise: 5%.

Parameter	True Value	Case 2N5	Case 3N5	Case 4N5	Case 5N5	Case 6N5
C_{lp}	-0.9782	-1.013	-1.102	-1.012	-1.006	-1.007
		(0.071)†	(0.072)	(0.072)	(0.076)	(0.076)
C_{lr}	0.4181	0.405	0.404	0.404	0.400	0.400
		(0.074)	(0.074)	(0.074)	(0.074)	(0.074)
$C_{l\beta}$	-0.1264	-0.118	-0.118	-0.110	-0.117	-0.109
		(0.008)	(0.008)	(0.007)	(0.008)	(0.007)
$C_{l\delta a}$	-0.2469	-0.240	-0.240	-0.240	-0.238	-0.238
		(0.023)	(0.024)	(0.024)	(0.025)	(0.025)
$C_{l\delta r}$	0.0465	0.043	0.043	0.043	0.043	0.043
		(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
C_{np}	-0.1153	-0.129	-0.130	-0.130	-0.136	-0.135
		(0.034)	(0.035)	(0.035)	(0.038)	(0.038)
C_{nr}	-0.4949	-0.588	-0.585	-0.586	-0.575	-0.576
		(0.061)	(0.060)	(0.060)	(0.059)	(0.059)
$C_{n\beta}$	0.2805	0.280	0.280	0.261	0.279	0.261
		(0.013)	(0.013)	(0.012)	(0.013)	(0.012)
$C_{n\delta a}$	0.00	0.001	0.001	0.001	0.001	0.001
		(0.009)	(0.009)	(0.009)	(0.011)	(0.011)
$C_{n\delta r}$	-0.1659	-0.171	-0.171	-0.171	-0.170	-0.170
		(0.011)	(0.010)	(0.011)	(0.011)	(0.011)
C_{yp}	0.3029	0.352	0.356	0.356	0.372	0.371
		(0.168)	(0.173)	(0.173)	(0.192)	(0.190)
C_{yr}	0.7273	0.600	0.595	0.595	0.580	0.581
		(0.090)	(0.092)	(0.092)	(0.094)	(0.094)
$C_{y\beta}$	-1.1328	-1.109	-1.109	-1.036	-1.108	-1.036
		(0.070)	(0.070)	(0.065)	(0.067)	(0.063)
$C_{y\delta a}$	0.0293	0.028	0.029	0.029	0.030	0.030
		(0.024)	(0.026)	(0.026)	(0.034)	(0.033)
$C_{y\delta r}$	0.1914	0.192	0.192	0.192	0.192	0.192
		(0.020)	(0.020)	(0.020)	(0.019)	(0.019)

† Sample standard deviation

Table 5.5 Effect of bias and scale factor on parameter estimates via the Zero method. Measurement noise: 5%.

Parameter	True Value	Case 2N5	Case 3N5	Case 4N5	Case 5N5	Case 6N5
C_{lp}	-0.9782	-1.164	-1.176	-1.175	-1.212	-1.210
		(0.060)†	(0.056)	(0.056)	(0.044)	(0.044)
C_{lr}	0.4181	0.509	0.520	0.520	0.563	0.560
		(0.021)	(0.021)	(0.021)	(0.021)	(0.021)
$C_{l\beta}$	-0.1264	-0.127	-0.129	-0.121	-0.136	-0.126
		(0.007)	(0.007)	(0.006)	(0.007)	(0.006)
$C_{l\delta a}$	-0.2469	-0.229	-0.232	-0.232	-0.242	-0.242
		(0.030)	(0.029)	(0.029)	(0.023)	(0.023)
$C_{l\delta r}$	0.0465	0.041	0.042	0.042	0.042	0.042
		(0.002)	(0.002)	(0.002)	(0.002)	(0.002)
C_{l0}	0.0009	-0.0011	-0.0005	-0.0005	0.0023	0.0020
C_{np}	-0.1153	-0.113	-0.102	-0.103	-0.058	-0.062
		(0.020)	(0.021)	(0.021)	(0.023)	(0.023)
C_{nr}	-0.4949	-0.702	-0.697	-0.698	-0.666	-0.670
		(0.017)	(0.017)	(0.017)	(0.017)	(0.017)
$C_{n\beta}$	0.2805	0.301	0.303	0.282	0.307	0.287
		(0.008)	(0.008)	(0.008)	(0.009)	(0.008)
$C_{n\delta a}$	0.00	0.005	0.008	0.008	0.020	0.019
		(0.011)	(0.013)	(0.013)	(0.017)	(0.017)
$C_{n\delta r}$	-0.1659	-0.177	-0.176	-0.176	-0.168	-0.169
		(0.012)	(0.013)	(0.013)	(0.017)	(0.017)
C_{n0}	0.0016	0.0022	0.0007	0.0008	-0.0053	-0.0048
C_{yp}	0.3029	0.251	0.195	0.199	-0.031	-0.013
		(0.057)	(0.054)	(0.054)	(0.039)	(0.040)
C_{yr}	0.7273	0.652	0.638	0.639	0.592	0.595
		(0.017)	(0.017)	(0.017)	(0.015)	(0.014)
$C_{y\beta}$	-1.1328	-1.226	-1.234	-1.152	-1.258	-1.175
		(0.051)	(0.051)	(0.048)	(0.049)	(0.046)
$C_{y\delta a}$	0.0293	0.041	0.034	0.035	-0.001	0.002
		(0.034)	(0.037)	(0.037)	(0.053)	(0.052)
$C_{y\delta r}$	0.1914	0.210	0.207	0.207	0.191	0.193
		(0.030)	(0.030)	(0.031)	(0.030)	(0.030)
C_{y0}	-0.0045	-0.0057	0.0005	0.0001	0.0257	0.0236

† Sample standard deviation

Table 5.6 Effect of bias and scale factor on parameter estimates via the Delta method. Measurement noise: 10%.

Parameter	True Value	Case 2N10	Case 3N10	Case 4N10	Case 5N10	Case 6N10
C_{lp}	-0.9782	-0.909	-0.903	-0.904	-0.883	-0.885
		(0.092)†	(0.093)	(0.093)	(0.095)	(0.095)
C_{lr}	0.4181	0.350	0.349	0.349	0.348	0.349
		(0.070)	(0.070)	(0.070)	(0.068)	(0.068)
$C_{l\beta}$	-0.1264	-0.102	-0.101	-0.095	-0.099	-0.092
		(0.012)	(0.012)	(0.011)	(0.013)	(0.013)
$C_{l\delta a}$	-0.2469	-0.218	-0.217	-0.217	-0.214	-0.214
		(0.028)	(0.029)	(0.029)	(0.029)	(0.029)
$C_{l\delta r}$	0.0465	0.038	0.038	0.038	0.039	0.039
		(0.004)	(0.005)	(0.005)	(0.006)	(0.006)
C_{np}	-0.1153	-0.132	-0.133	-0.133	-0.134	-0.134
		(0.036)	(0.037)	(0.037)	(0.042)	(0.042)
C_{nr}	-0.4949	-0.533	-0.528	-0.528	-0.514	-0.515
		(0.061)	(0.063)	(0.063)	(0.067)	(0.067)
$C_{n\beta}$	0.2805	0.280	0.280	0.261	0.280	0.261
		(0.013)	(0.013)	(0.012)	(0.014)	(0.013)
$C_{n\delta a}$	0.00	0.005	0.005	0.005	0.006	0.006
		(0.015)	(0.016)	(0.016)	(0.020)	(0.020)
$C_{n\delta r}$	-0.1659	-0.167	-0.167	-0.167	-0.166	-0.166
		(0.011)	(0.011)	(0.011)	(0.011)	(0.011)
C_{yp}	0.3029	0.358	0.365	0.364	0.388	0.386
		(0.190)	(0.198)	(0.197)	(0.228)	(0.226)
C_{yr}	0.7273	0.531	0.527	0.527	0.514	0.514
		(0.089)	(0.093)	(0.092)	(0.107)	(0.106)
$C_{y\beta}$	-1.1328	-1.115	-1.115	-1.042	-1.113	-1.040
		(0.069)	(0.069)	(0.064)	(0.065)	(0.061)
$C_{y\delta a}$	0.0293	0.032	0.033	0.033	0.040	0.040
		(0.058)	(0.063)	(0.062)	(0.079)	(0.078)
$C_{y\delta r}$	0.1914	0.190	0.190	0.190	0.192	0.192
		(0.018)	(0.018)	(0.018)	(0.017)	(0.018)

† Sample standard deviation

Table 5.7 Effect of bias and scale factor on parameter estimates via the Zero method. Measurement noise: 10%.

Parameter	True Value	Case 2N10	Case 3N10	Case 4N10	Case 5N10	Case 6N10
C_{lp}	-0.9782	-1.132	-1.144	-1.143	-1.184	-1.181
		(0.064)†	(0.062)	(0.062)	(0.052)	(0.053)
C_{lr}	0.4181	0.444	0.453	0.453	0.495	0.491
		(0.021)	(0.021)	(0.021)	(0.023)	(0.023)
$C_{l\beta}$	-0.1264	-0.115	-0.117	-0.109	-0.126	-0.117
		(0.009)	(0.009)	(0.008)	(0.010)	(0.009)
$C_{l\delta a}$	-0.2469	-0.208	-0.210	-0.210	-0.216	-0.216
		(0.035)	(0.035)	(0.035)	(0.032)	(0.032)
$C_{l\delta r}$	0.0465	0.045	0.046	0.046	0.049	0.048
		(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
C_{l0}	0.0009	-0.0023	-0.0018	-0.0018	0.0007	0.0005
C_{np}	-0.1153	-0.085	-0.073	-0.074	-0.026	-0.030
		(0.024)	(0.025)	(0.025)	(0.026)	(0.026)
C_{nr}	-0.4949	-0.668	-0.661	-0.661	-0.626	-0.629
		(0.016)	(0.016)	(0.016)	(0.016)	(0.016)
$C_{n\beta}$	0.2805	0.311	0.313	0.292	0.320	0.298
		(0.009)	(0.009)	(0.009)	(0.009)	(0.009)
$C_{n\delta a}$	0.00	0.011	0.015	0.014	0.031	0.030
		(0.018)	(0.020)	(0.020)	(0.028)	(0.028)
$C_{n\delta r}$	-0.1659	-0.174	-0.173	-0.173	-0.165	-0.166
		(0.011)	(0.012)	(0.012)	(0.016)	(0.016)
C_{n0}	0.0016	0.0023	0.0007	0.0008	-0.0055	-0.0050
C_{yp}	0.3029	0.218	0.167	0.170	-0.028	-0.013
		(0.040)	(0.035)	(0.036)	(0.015)	(0.017)
C_{yr}	0.7273	0.588	0.584	0.584	0.600	0.594
		(0.015)	(0.015)	(0.015)	(0.023)	(0.022)
$C_{y\beta}$	-1.1328	-1.232	-1.241	-1.159	-1.273	-1.187
		(0.052)	(0.053)	(0.049)	(0.056)	(0.052)
$C_{y\delta a}$	0.0293	0.059	0.054	0.054	0.023	0.026
		(0.067)	(0.074)	(0.073)	(0.096)	(0.095)
$C_{y\delta r}$	0.1914	0.207	0.205	0.206	0.198	0.199
		(0.027)	(0.027)	(0.027)	(0.025)	(0.025)
C_{y0}	-0.0045	-0.0078	-0.0018	-0.0022	0.0235	0.0214

† Sample standard deviation

Table 5.8 Effect of including free response for parameter estimation via the Delta method.

Parameter	True Value	Case 1	Case 1 + Free Response	Case 6N5	Case 6N5 + Free Response
C_{lp}	-0.9782	-1.063	-0.951	-1.007	-0.850
		(0.051)†	(0.018)	(0.076)	(0.038)
C_{lr}	0.4181	0.430	0.440	0.400	0.497
		(0.069)	(0.010)	(0.074)	(0.020)
$C_{l\beta}$	-0.1264	-0.125	-0.125	-0.109	-0.109
		(0.008)	(0.002)	(0.007)	(0.004)
$C_{l\delta a}$	-0.2469	-0.251	-0.243	-0.239	-0.222
		(0.014)	(0.004)	(0.025)	(0.005)
$C_{l\delta r}$	0.0465	0.042	0.049	0.043	0.057
		(0.004)	(0.001)	(0.005)	(0.003)
C_{np}	-0.1153	-0.108	-0.097	-0.135	-0.153
		(0.025)	(0.019)	(0.037)	(0.028)
C_{nr}	-0.4949	-0.622	-0.472	-0.576	-0.510
		(0.058)	(0.035)	(0.059)	(0.023)
$C_{n\beta}$	0.2805	0.277	0.280	0.261	0.264
		(0.013)	(0.007)	(0.012)	(0.009)
$C_{n\delta a}$	0.00	-0.002	0.003	0.001	-0.001
		(0.006)	(0.004)	(0.011)	(0.006)
$C_{n\delta r}$	-0.1659	-0.173	-0.166	-0.170	-0.169
		(0.009)	(0.005)	(0.010)	(0.004)
C_{yp}	0.3029	0.291	0.310	0.371	0.417
		(0.145)	(0.072)	(0.190)	(0.021)
C_{yr}	0.7273	0.667	0.779	0.581	0.919
		(0.078)	(0.087)	(0.094)	(0.067)
$C_{y\beta}$	-1.1328	-1.098	-1.147	-1.036	-1.067
		(0.075)	(0.025)	(0.063)	(0.035)
$C_{y\delta a}$	0.0293	0.025	0.034	0.030	0.044
		(0.015)	(0.007)	(0.033)	(0.005)
$C_{y\delta r}$	0.1914	0.194	0.205	0.192	0.216
		(0.021)	(0.007)	(0.019)	(0.008)

† Sample standard deviation

Table 5.9 Effect of estimated parameters via the Delta and the Zero methods, considering only free response (with no measurement errors).

Parameter	True Value	Delta method	Zero method
C_{lp}	-0.9782	-0.0945	-0.0882
		(0.038)†	(0.105)
C_{lr}	0.4181	0.6247	0.6471
		(0.018)	(0.043)
$C_{l\beta}$	-0.1264	-0.0416	-0.0615
		(0.004)	(0.020)
C_{l0}	0.0009	-	0.0002
C_{np}	-0.1153	-1.3340	-1.0536
		(0.090)	(0.074)
C_{nr}	-0.4949	-0.8825	-0.6487
		(0.067)	(0.210)
$C_{n\beta}$	0.2805	0.1769	0.2321
		(0.010)	(0.083)
C_{n0}	0.0016	-	0.0021
C_{yp}	0.3029	2.8268	2.6033
		(0.300)	(0.197)
C_{yr}	0.7273	1.2563	1.305
		(0.339)	(0.796)
$C_{y\beta}$	-1.1328	-0.8550	-0.8692
		(0.080)	(0.223)
C_{y0}	-0.0045	-	-0.0059

† Sample standard deviation

Table 5.10 Parameter estimates from real flight data via the Delta and the Zero methods.

Parameter	ML	Ref. 13	Case R1 Delta method	Case R2 Delta method	Case R1 Zero method	Case R2 Zero method
C_{lp}	-0.9782	-0.866	-0.943	-0.899	-0.841	-0.916
	(0.20)*	(0.013)†	(0.030)	(0.068)	(0.027)	(0.035)
C_{lr}	0.4181	0.594	0.509	0.666	0.437	0.608
	(0.43)	(0.028)	(0.060)	(0.100)	(0.101)	(0.081)
$C_{l\beta}$	-0.1264	-0.110	-0.102	-0.136	-0.132	-0.159
	(0.39)	(0.003)	(0.014)	(0.016)	(0.006)	(0.006)
$C_{l\delta a}$	-0.2469	-0.209	-0.237	-0.224	-0.222	-0.220
	(0.20)	(0.004)	(0.020)	(0.031)	(0.007)	(0.008)
$C_{l\delta r}$	0.0465	0.027	0.016	0.034	-0.028	0.016
	(0.90)	(0.007)	(0.026)	(0.016)	(0.022)	(0.020)
C_{l0}	-	-	-	-	0.0010	0.0008
C_{np}	-0.1153	-0.068	-0.101	-0.158	-0.099	-0.160
	(0.73)	(0.003)	(0.032)	(0.031)	(0.018)	(0.011)
C_{nr}	-0.4949	-0.864	-0.544	-0.567	-0.679	-0.497
	(0.56)	(0.026)	(0.087)	(0.055)	(0.053)	(0.054)
$C_{n\beta}$	0.2805	0.264	0.273	0.237	0.261	0.242
	(0.07)	(0.005)	(0.009)	(0.013)	(0.011)	(0.012)
$C_{n\delta a}$	0.0	-	0.014	-0.007	0.011	-0.013
			(0.003)	(0.010)	(0.007)	(0.004)
$C_{n\delta r}$	-0.1659	-0.169	-0.165	-0.180	-0.229	-0.224
	(0.12)	(0.004)	(0.013)	(0.010)	(0.007)	(0.010)
C_{n0}	-	-	-	-	0.0025	0.0024
C_{yp}	0.3029	0.111	0.304	0.266	0.362	0.410
	(5.18)	(0.025)	(0.065)	(0.060)	(0.014)	(0.018)
C_{yr}	0.7273	1.146	1.810	1.950	1.322	1.109
	(2.32)	(0.603)	(0.188)	(0.125)	(0.494)	(0.119)
$C_{y\beta}$	-1.1328	-1.083	-0.996	-1.153	-0.994	-1.089
	(0.33)	(0.049)	(0.032)	(0.157)	(0.060)	(0.089)
$C_{y\delta a}$	0.0293	0.129	0.116	0.117	0.041	0.069
	(13.5)	(0.011)	(0.045)	(0.047)	(0.016)	(0.004)
$C_{y\delta r}$	0.1914	0.095	0.009	0.124	0.014	0.031
	(2.02)	(0.008)	(0.053)	(0.069)	(0.012)	(0.021)
C_{y0}	-	-	-	-	0.0030	-0.0012

⋆ Standard deviation in percent as in Ref. 15

† Sample standard deviation

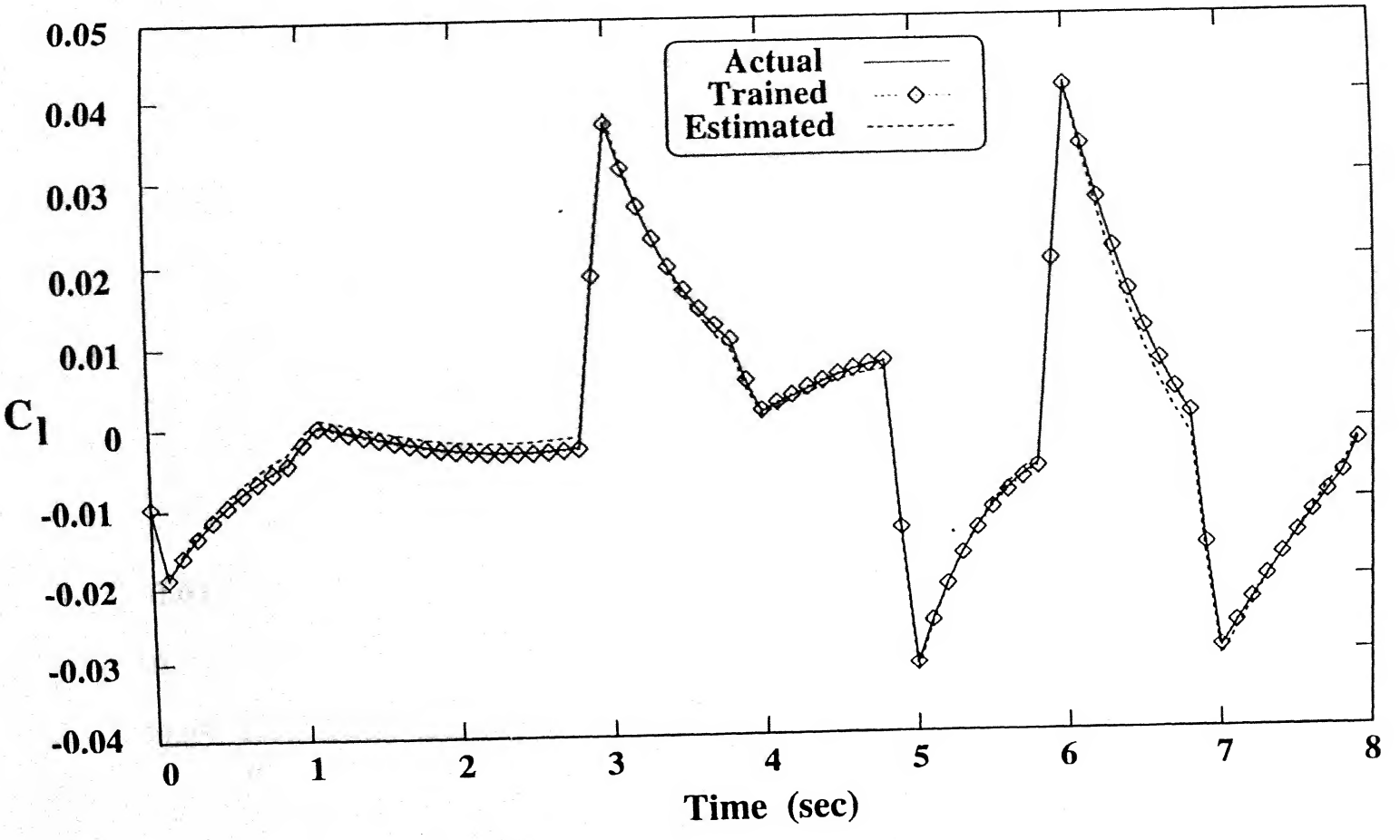


Fig. 5.1 Comparison of total aerodynamic coefficients for simulated flight data (case 1)

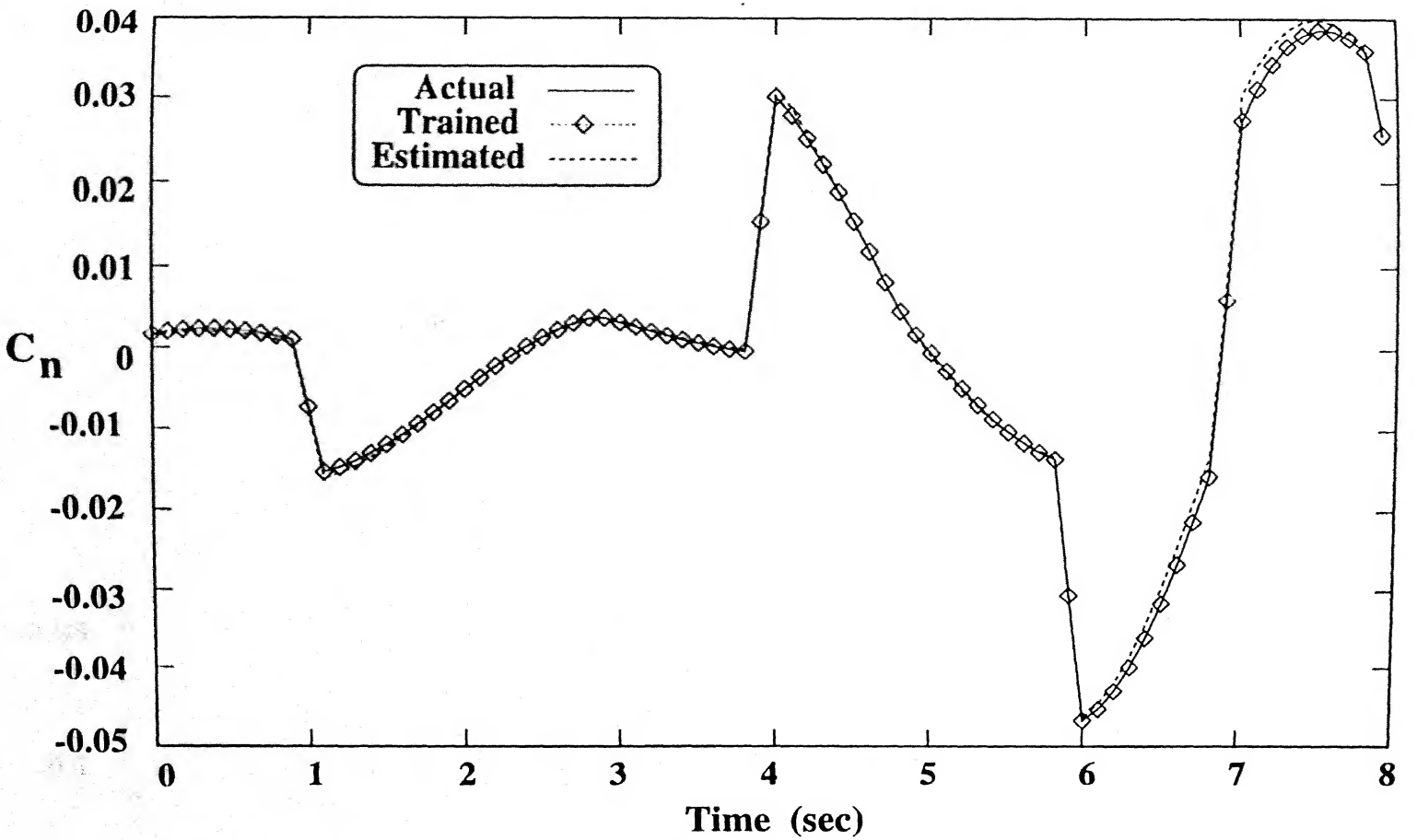


Fig. 5.1 Continued.

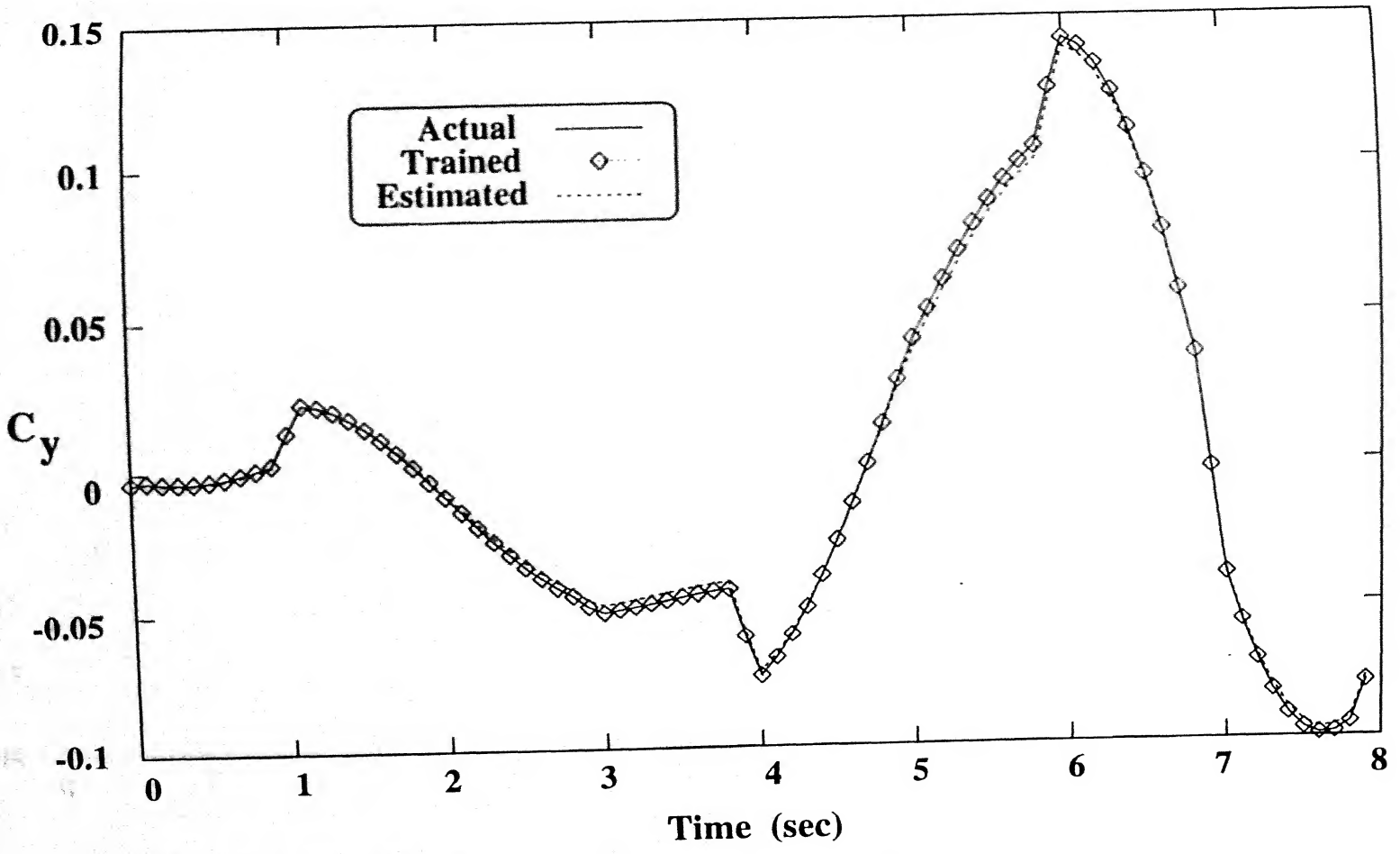


Fig. 5.1 Concluded.

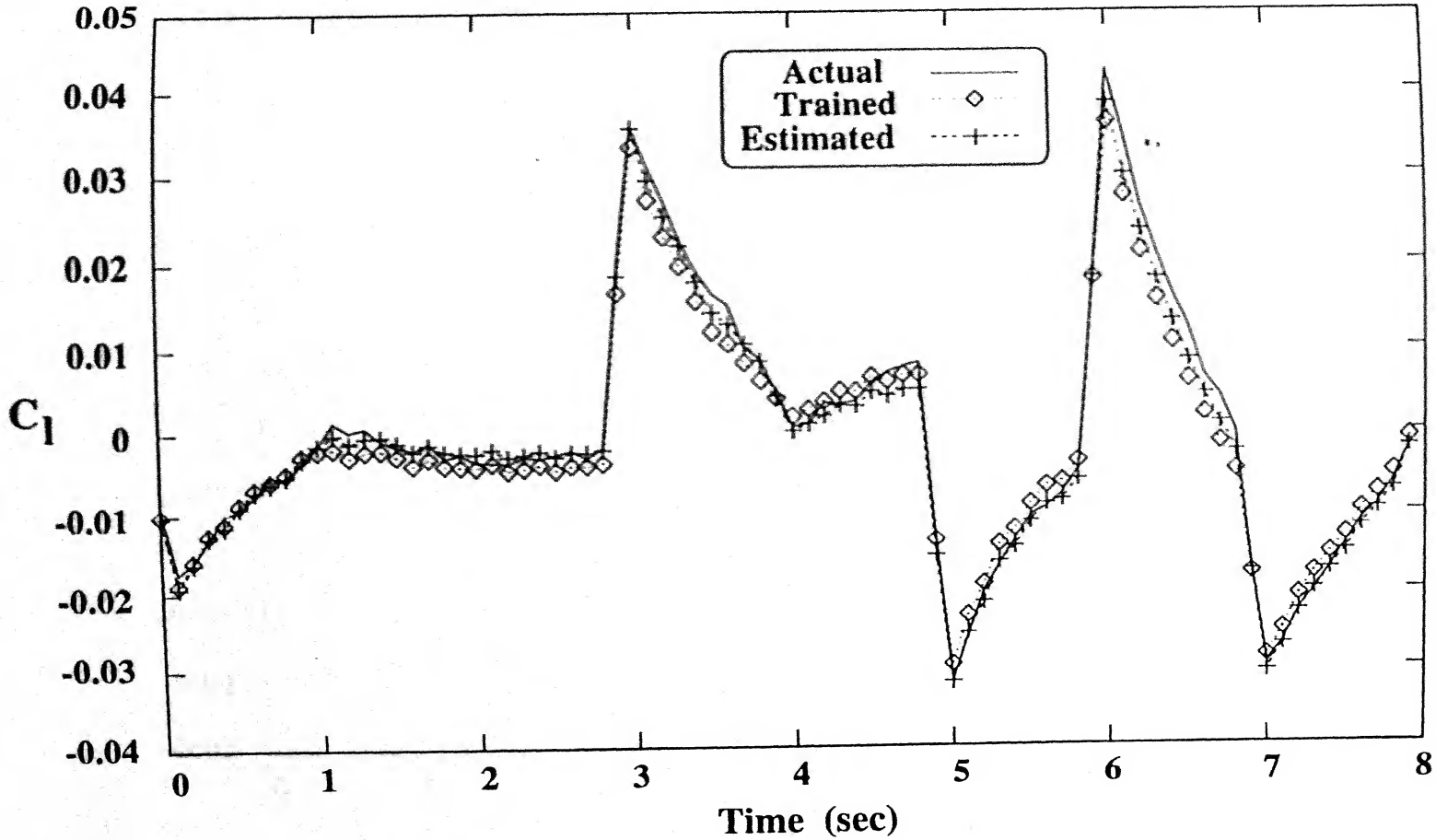


Fig. 5.2 Comparison of total aerodynamic coefficients for simulated flight data (case 6N5)

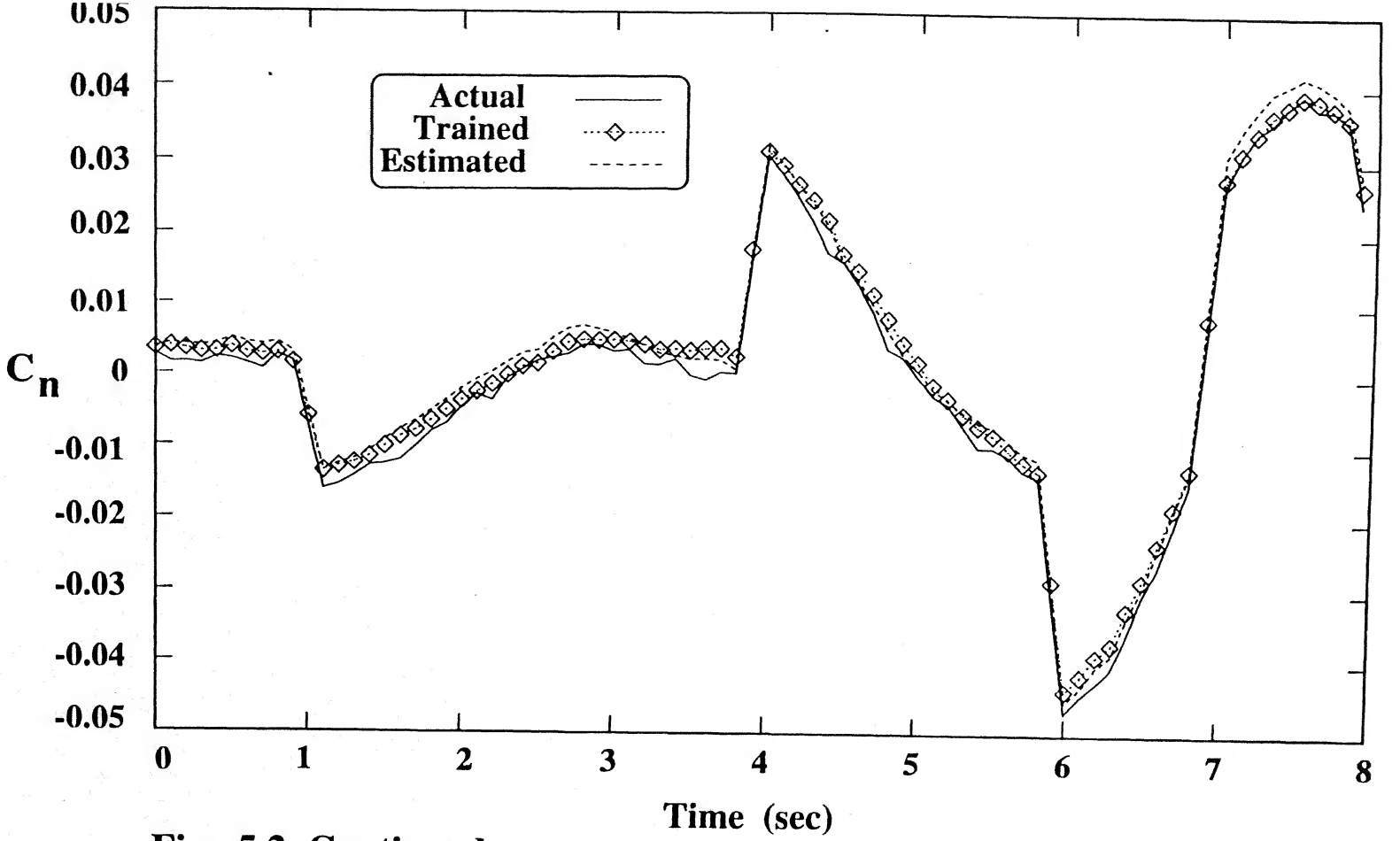


Fig. 5.2 Continued.

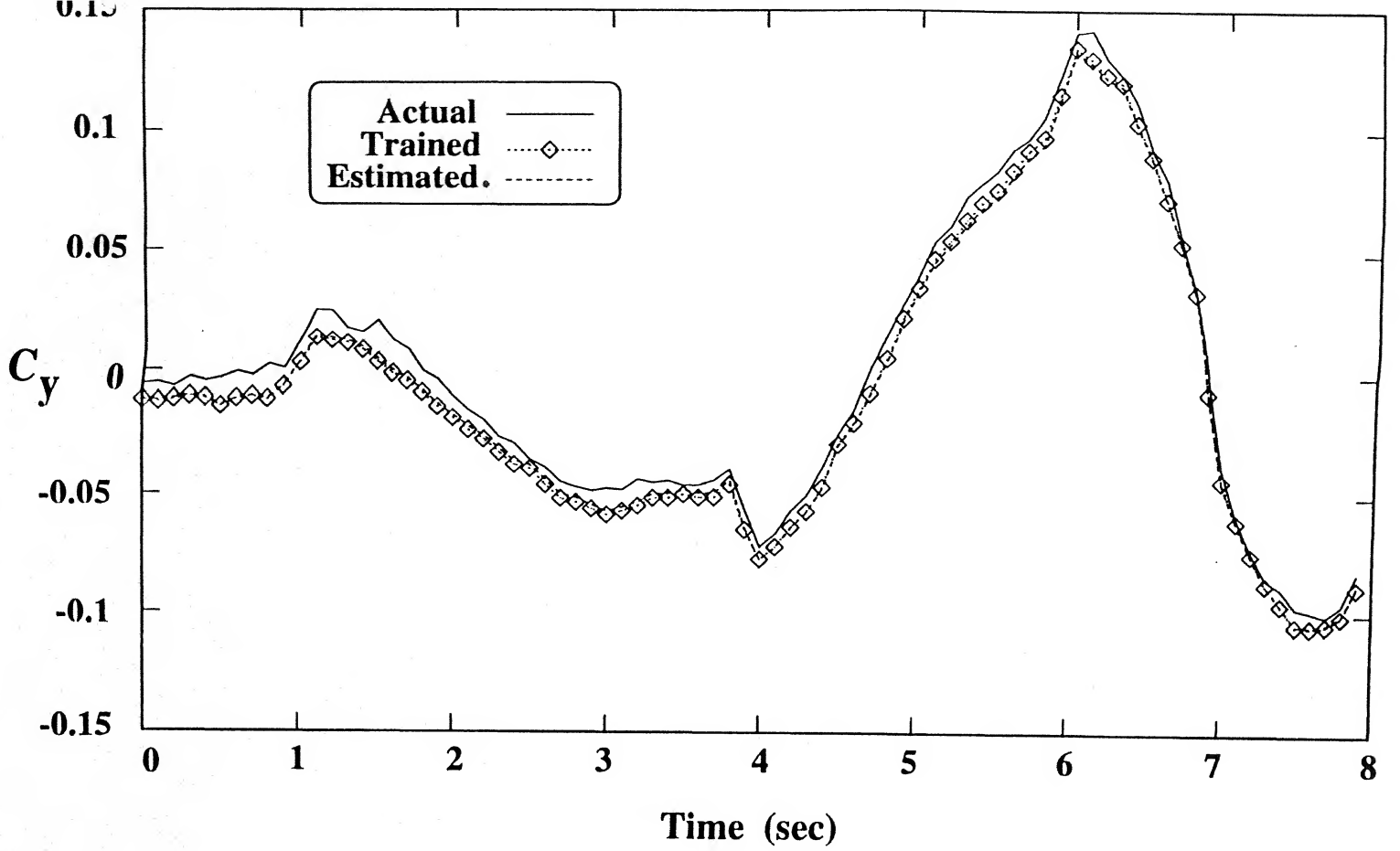


Fig. 5.2 Concluded.

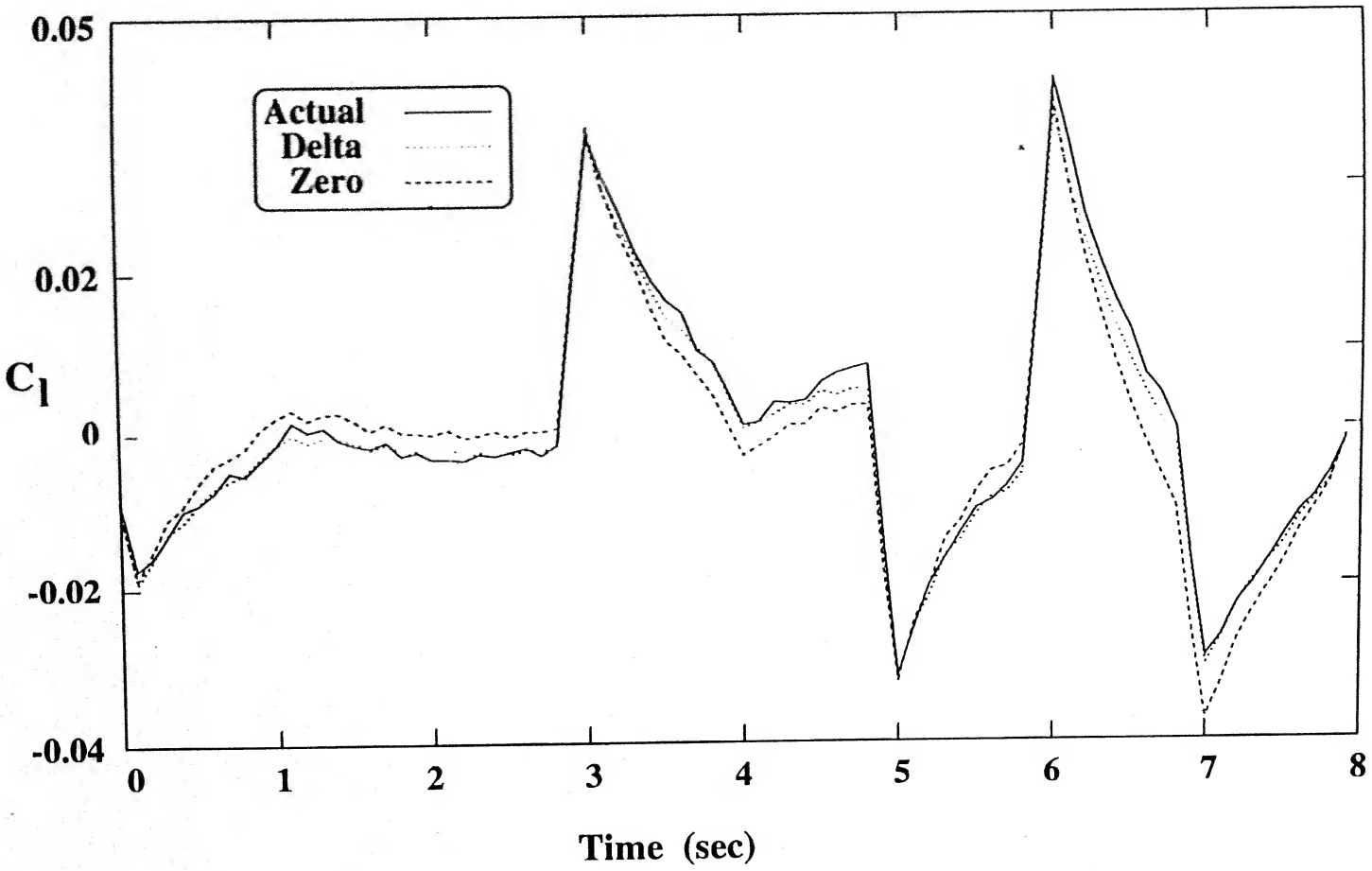


Fig. 5.3 Comparison of total aerodynamic coefficients for simulated flight data (case 6N5) via the Delta and the Zero methods.

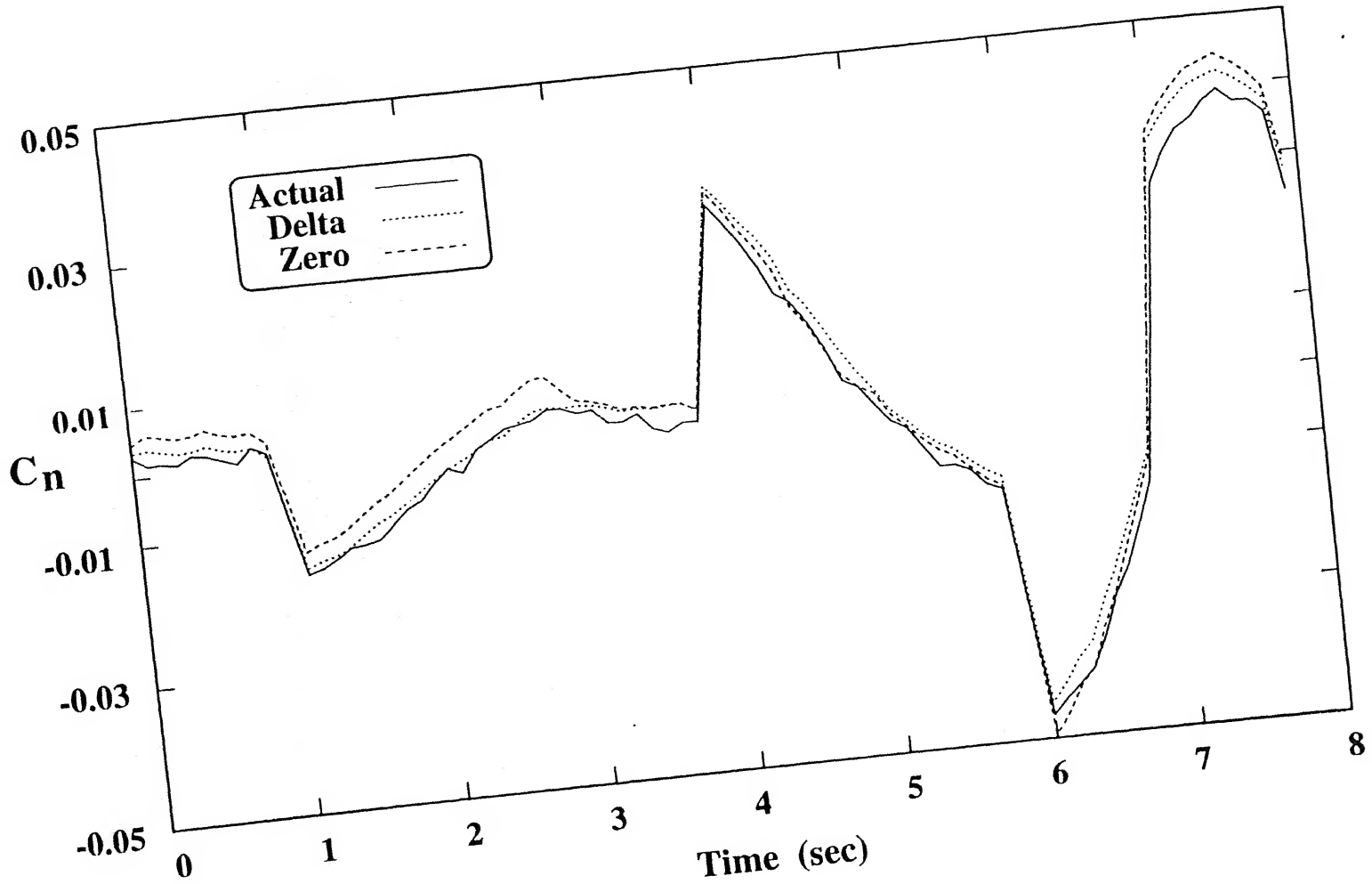


Fig. 5.3 Continued.

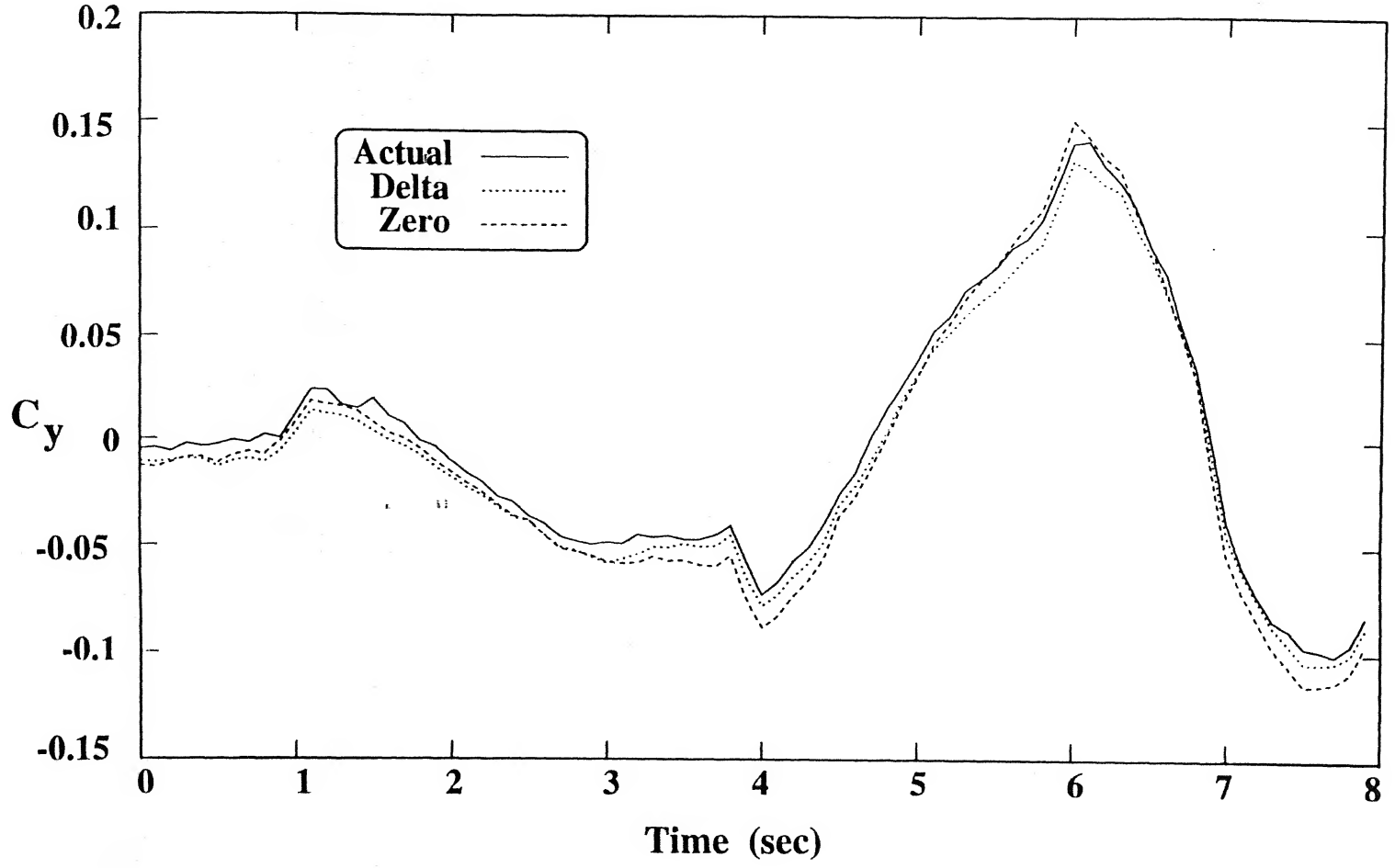


Fig. 5.3 Concluded.

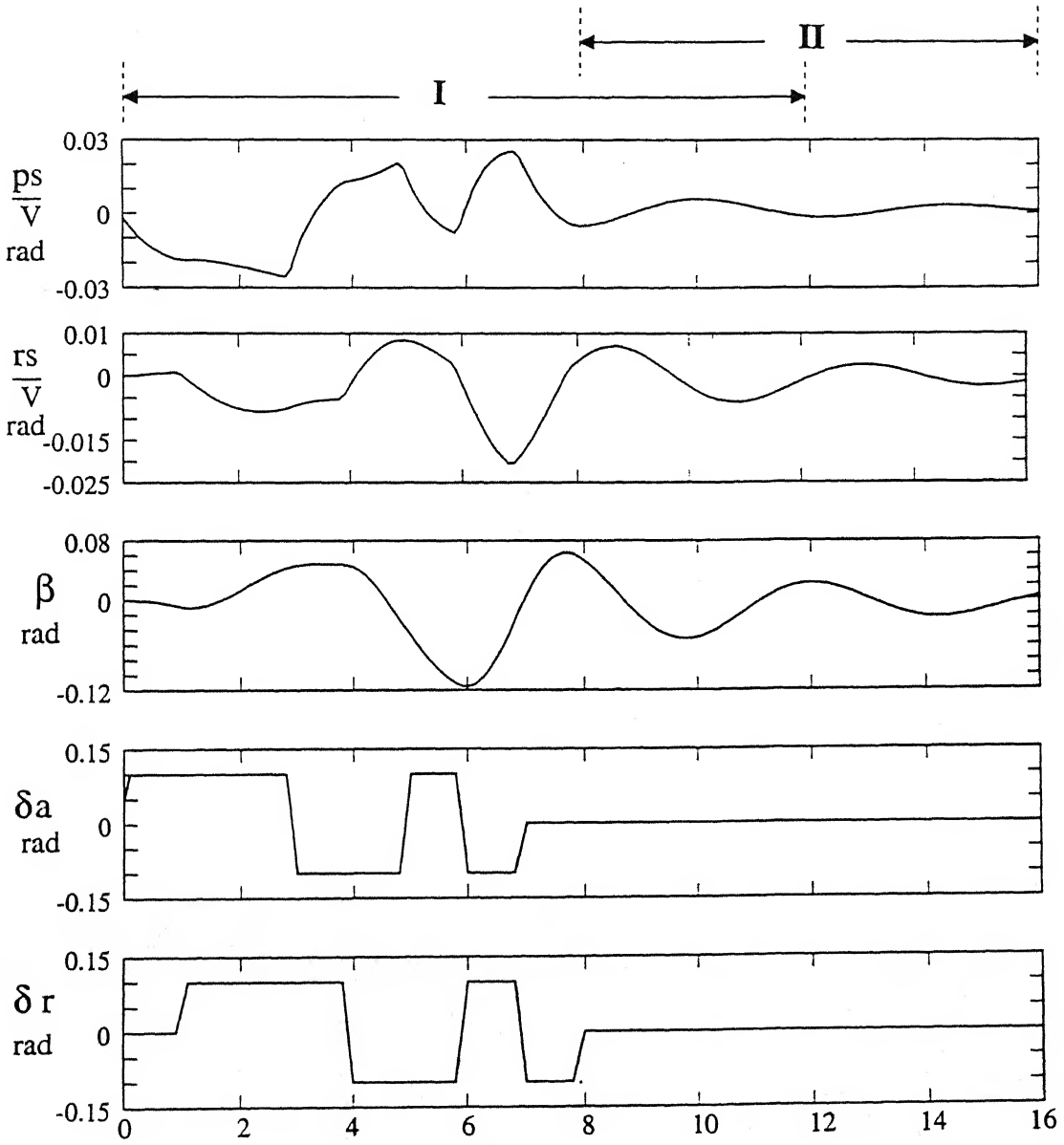
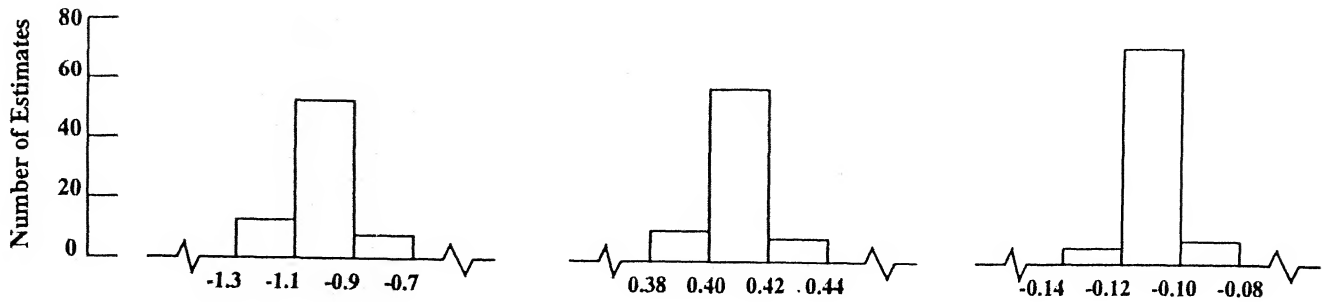
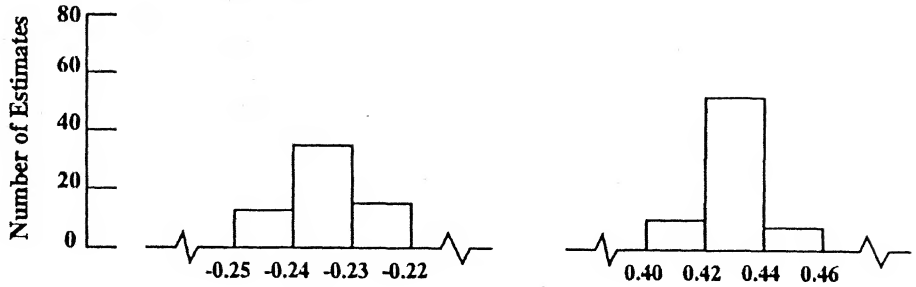


Fig. 5.4 Effect of including free response (8 sec) to forced response signal.

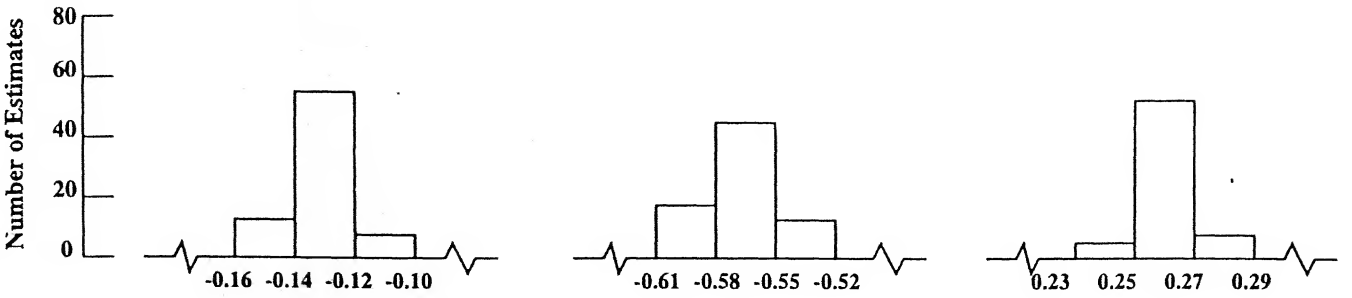


Parameter:	C_{lp}	C_{lr}	$C_{l\beta}$
True Value:	-0.9782	0.4181	-0.1264
Estimated Value:	-1.007 (0.076)	0.400 (0.074)	-0.109 (0.007)

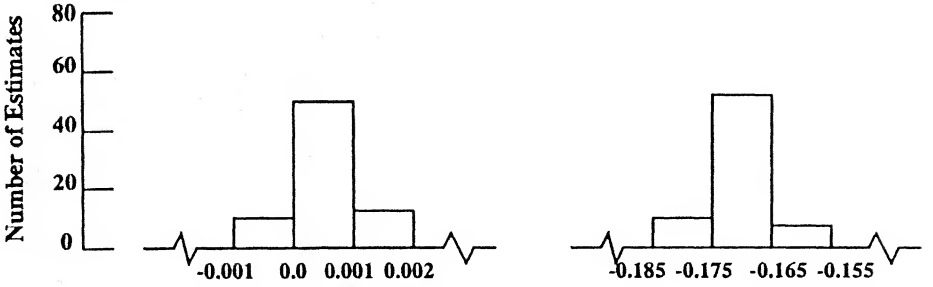


Parameter:	$C_{l\delta a}$	$C_{l\delta r}$
True Value:	-0.2469	0.0465
Estimated Value:	-0.238 (0.025)	0.043 (0.005)

Fig. 5.5 Histograms for C_l, C_n and C_y derivatives (case 6N5)via the Delta method.

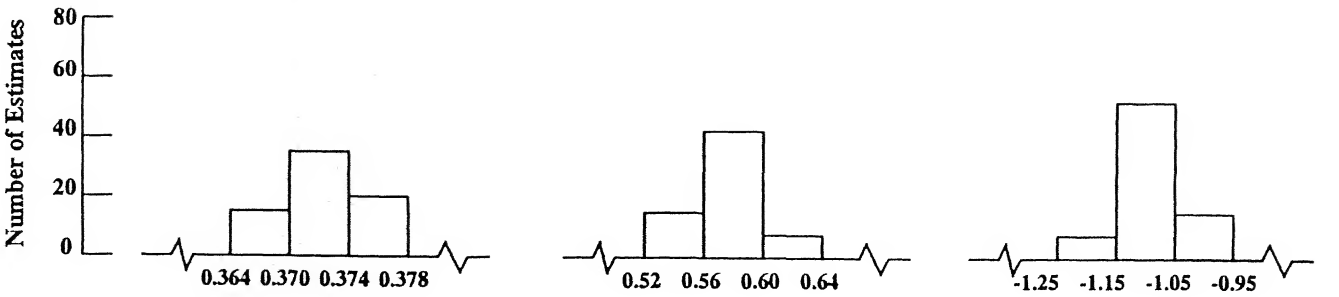


Parameter:	C_{np}	C_{nr}	$C_{n\beta}$
True Value:	-0.1153	-0.4949	0.2805
Estimated Value:	-0.135 (0.038)	-0.576 (0.059)	0.261 (0.012)

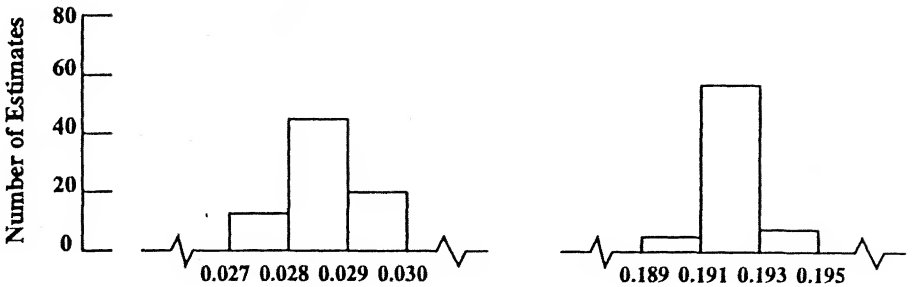


Parameter:	$C_{n\delta a}$	$C_{n\delta r}$
True Value:	0.0	-0.1659
Estimated Value:	0.001 (0.011)	-0.170 (0.011)

Fig. 5.5 Continued.



Parameter:	C_{yp}	C_{yr}	$C_{y\beta}$
True Value:	0.3029	0.7273	-1.1328
Estimated Value:	0.371 (0.190)	0.581 (0.094)	-1.036 (0.063)



Parameter:	$C_{y\delta a}$	$C_{y\delta r}$
True Value:	0.0293	0.1914
Estimated Value:	0.029 (0.032)	0.192 (0.019)

Fig. 5.5 Concluded.

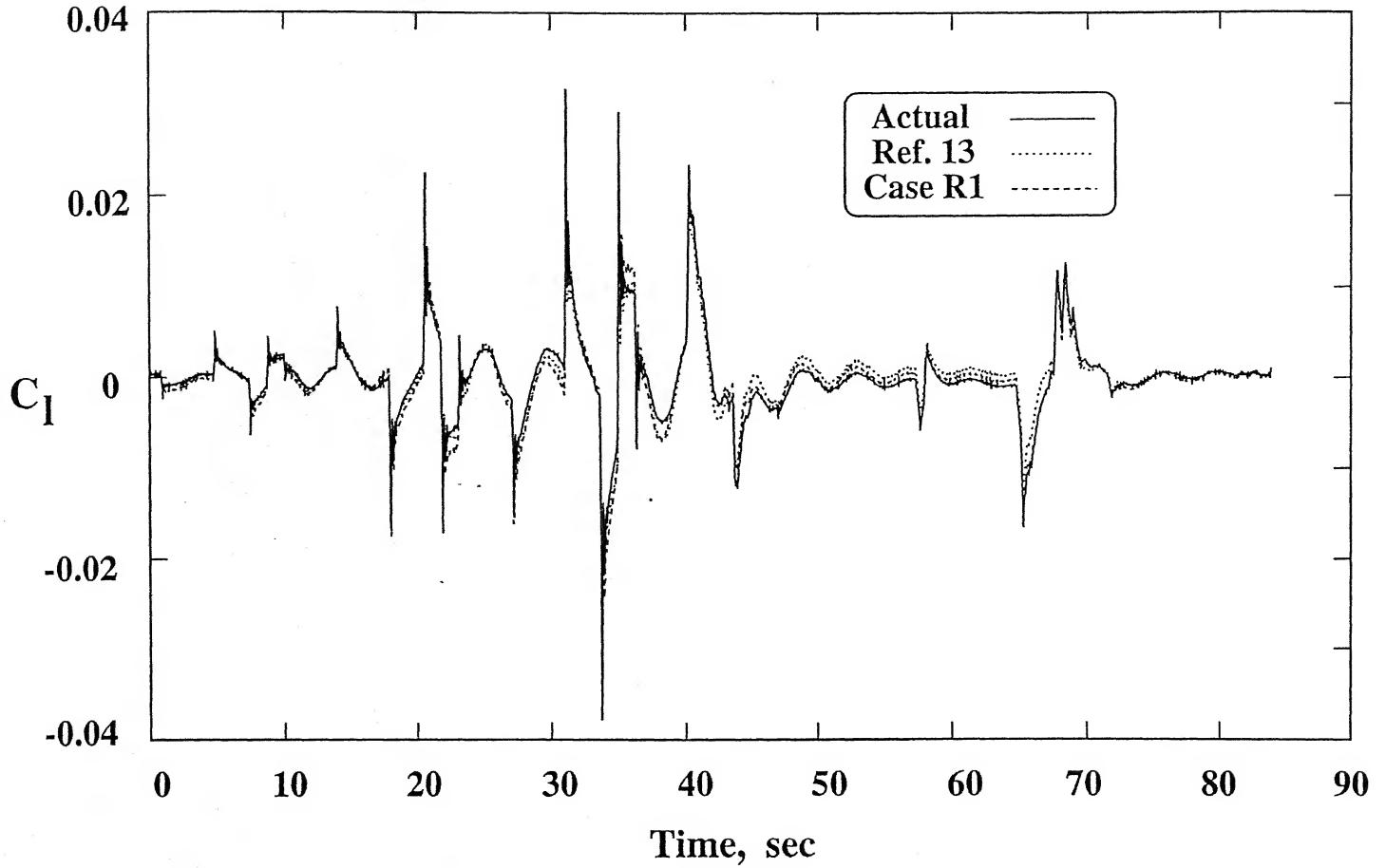


Fig. 5.6 Comparison of total aerodynamic coefficients for real flight data (case R1)

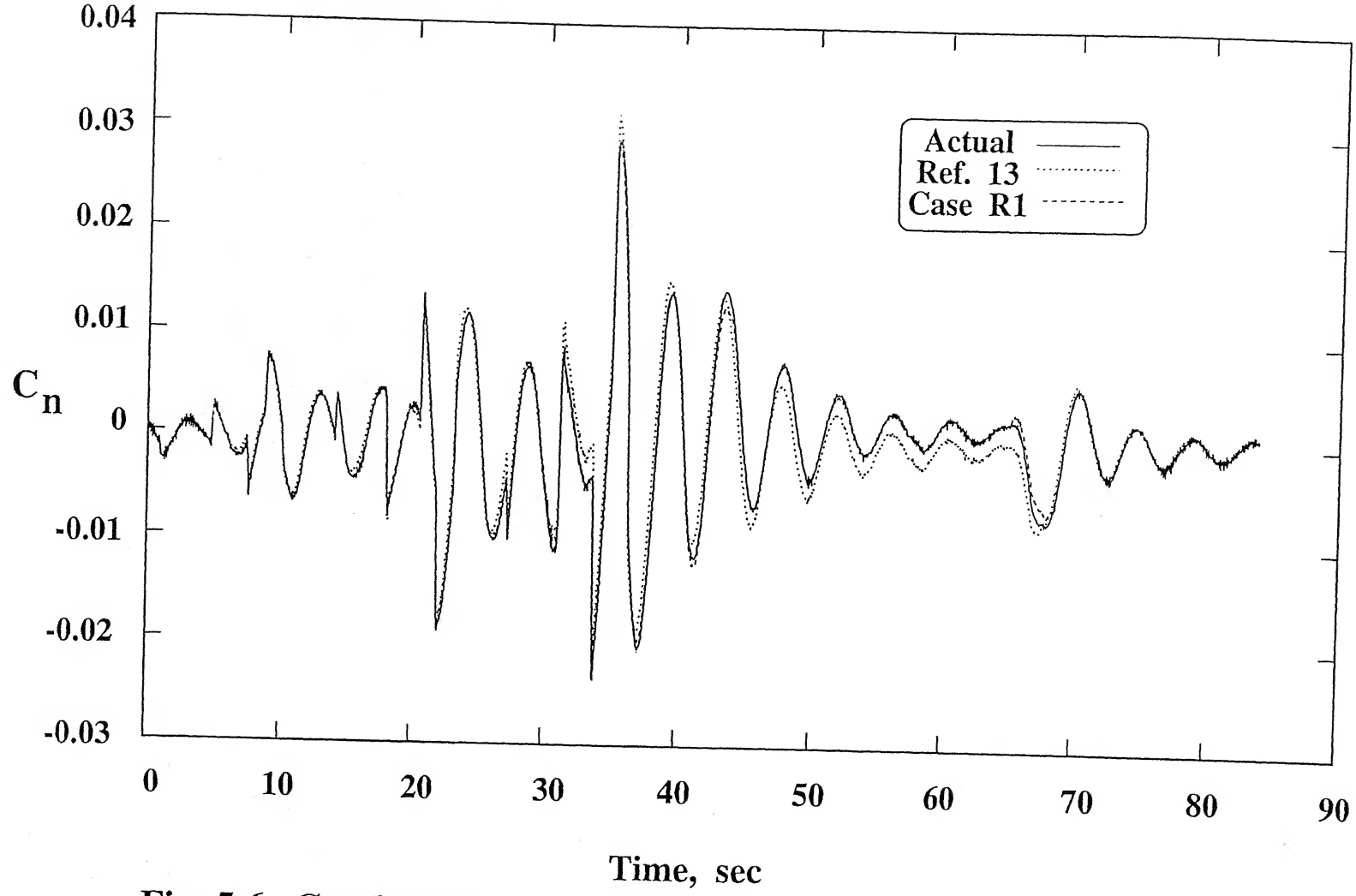


Fig. 5.6 Continued.

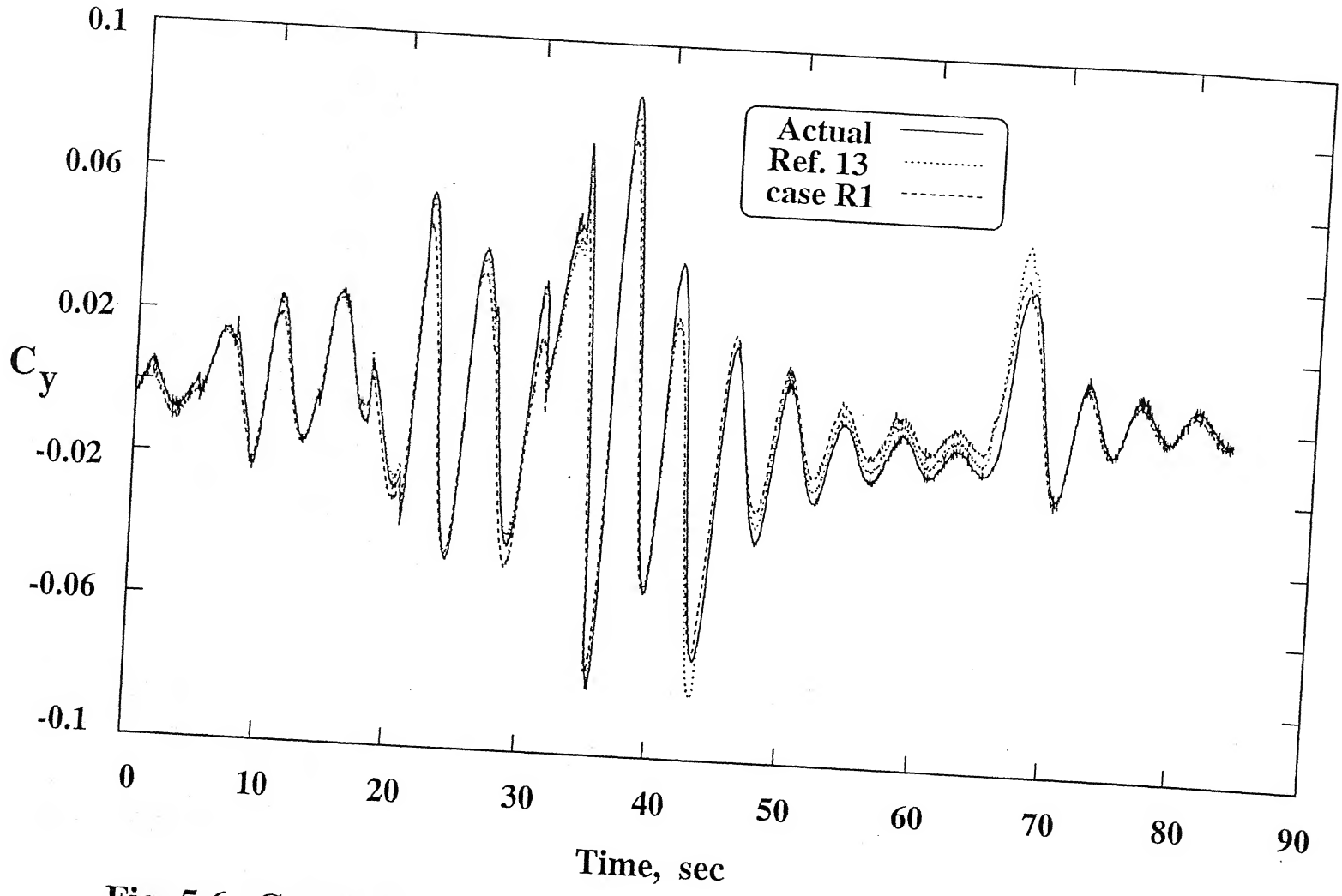


Fig. 5.6 Concluded.

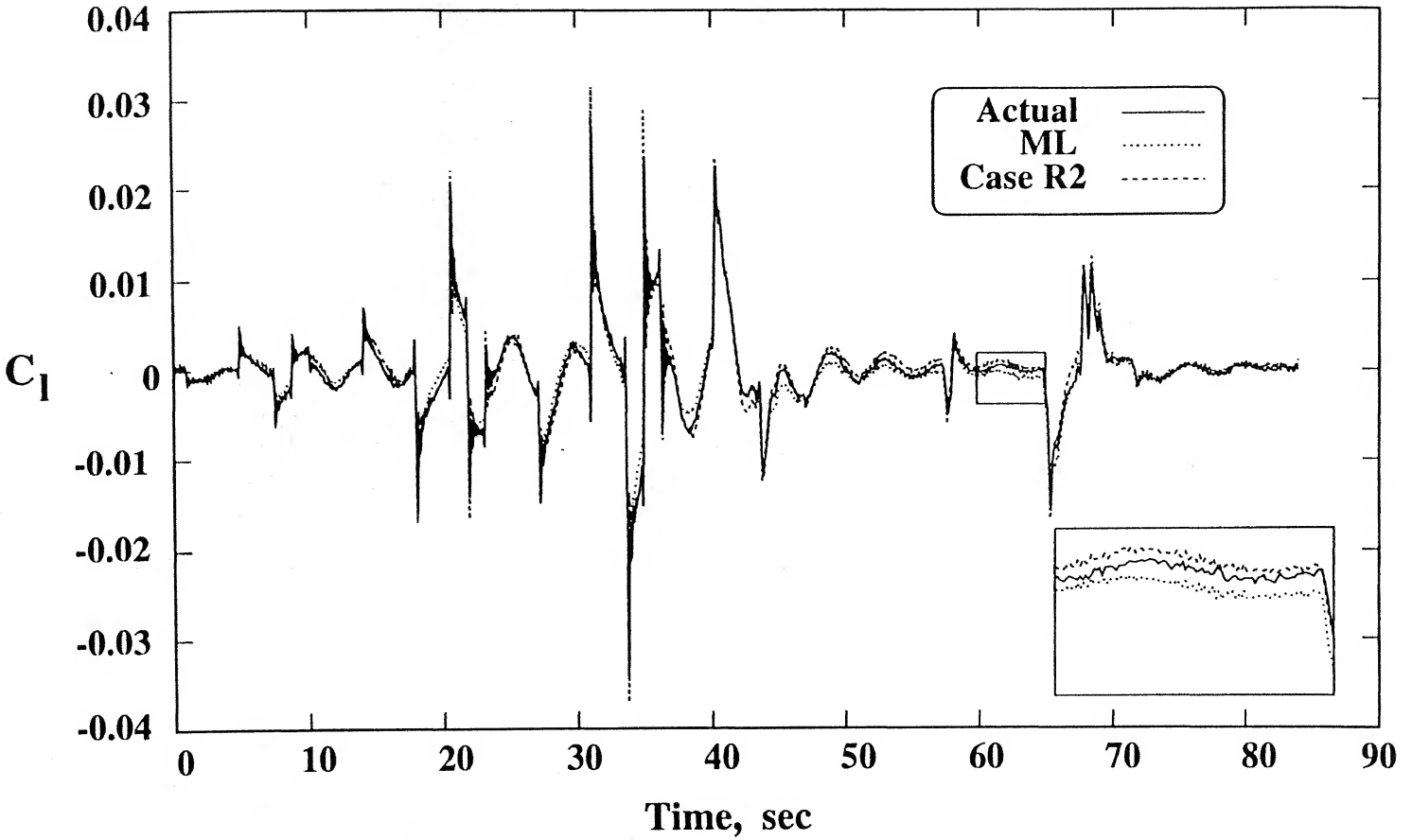


Fig. 5.7 Comparison of total aerodynamic coefficients for real flight data (case R2)

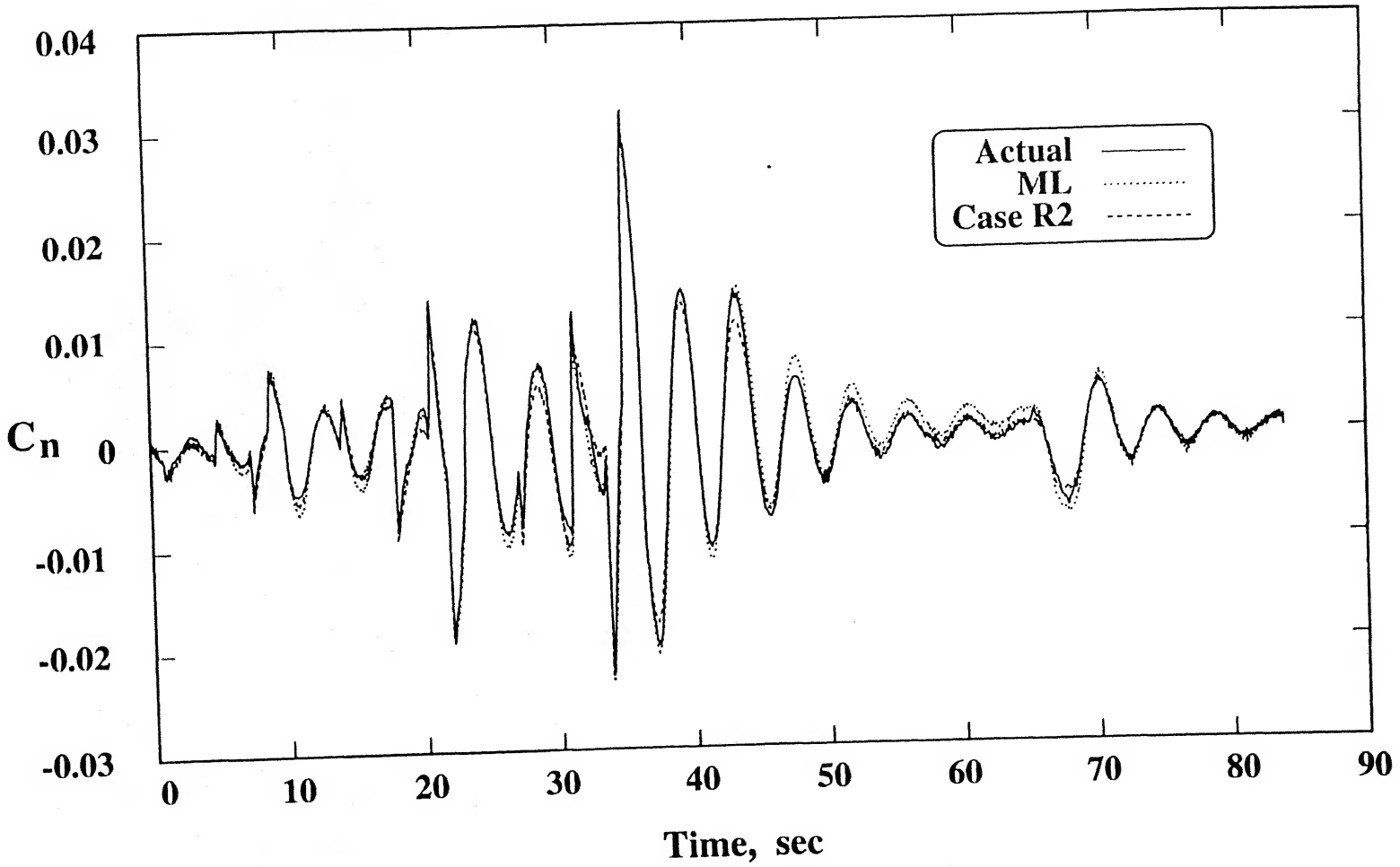


Fig. 5.7 Continued.

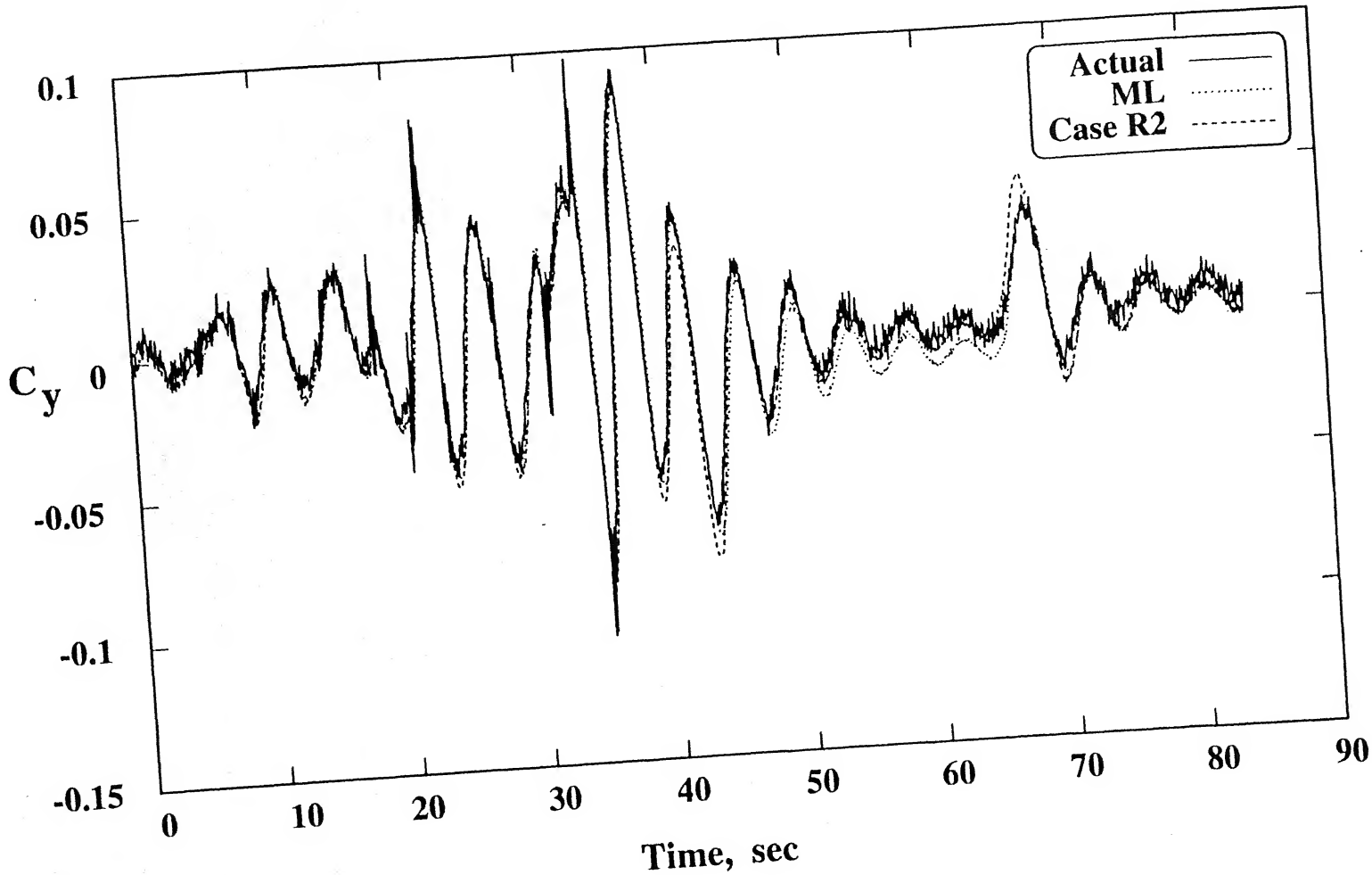


Fig. 5.7 Concluded.

Chapter 6

Conclusions

Applicability of the recently proposed Delta and Zero methods[12] for extracting aircraft parameters from simulated and real flight data has been demonstrated for the flight data containing unknown bias, scale factor and measurement noise. The recorded flight data is required to be corrected only for the time shift errors before applying the methods. The superiority of the Delta method over the Zero method, when used for estimating parameters from recorded data that contain measurement errors, has been demonstrated. The presence of bias and scale factor in recorded signals is shown to marginally affect the estimated parameters via the Delta method.The FFNN is able to establish a mapping between the network inputs of recorded motion and control variables and the network outputs computed using the recorded motion variables. It is conjectured that, in spite of the fact that the input-output mapping uses the recorded data directly (without corrections for bias and scale factor), the generalization properties of the FFNN enable it to correctly predict the modified output for the perturbed network input, and thus leading to good estimates of parameters via the Delta method.[12]

The results of the present study enhance the scope of the proposed methods[12]: the recorded data can be used directly without worrying about measurement errors (expect for the time shift). Furthermore, the concatenation of flight segments can be used to increase the information content of the

data without any requirement of estimating or matching the initial conditions of each segment. The results also help in exploring the potential of the proposed methods for on-line aerodynamic modelling and parameter estimation. For on-line parameter estimation, only the coefficient network needs to be carried on-board. The coefficients can be estimated using the aircraft sensors. Thus, such a network could quickly learn on-line to map the current motion and control inputs to the total coefficients, and yield parameter estimates for on-line use.

Bibliography

[1] Seckel, E., and Morris, J., "The Stability Derivatives of the Navion Aircraft Estimated by Various Methods and Derived from Flight Test Data," FAA-RD-71-6.

[2] Seckel, E., "Stability and control of Airplanes and helicopters," Academic Press; New York, 1964.

[3] Ellision, D. E., "USAF Stability and Control Hand Book (DATCOM)," Wright Patherson Airforce base, OHIO, August. 1968.

[4] Maine, R. E., and Illiff, K. W., " Identification of Dynamic Systems: Theory and Formulations," NASA RP 1138, Feb. 1985.

[5] Maine, R. E., and Illiff, K. W., "Application of Parameter Estimation to aircraft Stability and Control-The output-error Approach," NASA RP 1168, Jan. 1986.

[6] Mackie, D. B., "A comparison of parameter Estimation Results from Flight Data Using Linear and Nonlinear Maximum Likelihood Methods," DFVLR-FB 84-06, Braunschweig, Germany, Dec. 1983.

[7] Hornik, K., Stinchcombe, M., and White, H., "Multi Layer Feed Forward Networks are Universal Approximators," Neural Networks, Vol. 2, No. 5, 1989, pp. 359-366.

[8] Hess, R. A., "On the Use of Back Propagation with Feed Forward Neural Networks for Aerodynamic Estimation problem," AIAA Paper 93-3638, Aug. 1993.

[9] Linse, D. J., and Stengel, R. F., "Identification of Aerodynamic Coefficients Using Computational Neural Networks," Journal of Guidance, Control, and Dynamics, Vol. 16, No. 6, 1993, pp. 1018-1025.

[10] Youseff, H. M., "Estimation of Aerodynamic Coefficients Using Neural Networks," AIAA Paper 93-3639, Aug. 1993.

[11] Basappa, and Jategaonkar, R. V., "Aspect of feed Forward Neural Network Modelling and Its Application to lateral-Directional Flight Data," DLR-IB 111-95/30, Braunschweig, Germany, Sept. 1995.

[12] Raisinghani, S. C., Ghosh, A. K., and Kalra, P. K., "Two New Techniques for Aircraft Parameter Estimation Using Neural Networks," The Aeronautical Journal, Vol. 102, No. 1011, 1998, pp. 25-29.

[13] Ghosh, A. K., Raisinghani, S. C., and Khubchandani, S., "Estimation of Aircraft lateral-Directional Parameters Using Neural Networks," Journal of Aircraft, Vol. 35, No. 6, 1998, pp. 876-881.

[14] Raisinghani, S. C., Ghosh, A. K. and Kalra, P. K., "Aircraft lateral-directional parameter estimation via Neural Networks", Proceedings of 49^{th} AGM of the Aeronautical Society of India, Jan. 16-17: 6.1-6.7. 1998.

[15] Jategaonkar, R. V., "Identification of the Aerodynamic model of the DLR Research Aircraft ATTAS from Flight Test Data," DLR-FB 90-40, Braunschweig, Germany, Sept. 1990.

[16] Rakesh Kumar., "Parameter Estimation From Flight Data Using Feed Forward Neutral Networks," M.Tech Thesis, Aerospace Engg. Department, I.I.T.-Kanpur, February, 96.